## E.G.S PILLAY ENGINEERING COLLEGE, NAGAPATTINAM DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Approved by AICTE, Affiliated to Anna University) (Accredited by NBA - CSE/EEE/MECH) (Accredited by NAAC with "A" Grade) (An ISO 9001:2008 Certified Autonomous Institution)

## 17GEX51 - PROGRAMMING IN C LABORATARY MANUAL



## PREPARED BY

J. NOORUL AMEEN M.E, MISTE, EMCAA, $\overline{\text { Ph.D., }}$ ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## E.G.S PILLAY ENGINEERING COLLEGE, NAGAPATTINAM

Mail id: noornilo@gmail.com
Website: profameencse.weebly.com

## SYLLABUS



Ex.No: 1.a

## DOCUMENT CREATION

## Date:

Aim

To create a document and manipulate text with scientific notations.

## Procedure

1. Select Start $\rightarrow$ Programs $\rightarrow$ MS-Office Aricrosoft Word
2. Now new document window will be opened
3. Type the contents within the document and Perform the operations

Like
a) Make some text bold type, underline \& make italic
b) Correct spellings \& save the document with suitable name
c) Change the text with different font types \& size

## (I) COMPUTER

## Definition

Computer is an electronic machine which performs a particular task.

## Parts of the Computer

It has three components. They are

| i. | CPU |
| :--- | :--- |
| ii. | Input unit and |
| iii. | Output unit. |

CPU is the central Processing Unit which consists of three other units like ALU, Memory unit and Control unit. ALU means Arithmetic and Logic Unit which performs all arithmetic and logical operations. Memory unit is to store all programs and data. Control unit controls the overall system.
(II) Fundamentals of Computing

Fundamentals of computing

## Fundamentals of computing

ABS TEXTILES,
45,Car Street,
Tiruchengode-637 204.
04288-233871.
4. Search for the word Computer and replace it with Computer System
5. Type the following at the end of the document:

$$
\begin{aligned}
& (A+B)^{2}=A^{2}+B^{2}+2 A B \\
& \mathrm{H}_{2}+\mathrm{O}_{2} \rightarrow \mathrm{H}_{2} \mathrm{O} \\
& \text { va+b=c }
\end{aligned}
$$

6. For searching operations press find button in home tools or press F5 key
7. Save the document as Filename.doc
8. Close the document

## RESULT

Thus the document has been created and manipulated text with scientific notations..

## Ex.No: 1.b PREPARATION OF AN ADVERTISEMENT USING MS - WORD

## Date:

## AIM:

To prepare an advertisement for a company with the following specifications using MSWORD.
a. Attractive Page border
b. Use at least one Clip art
c. Design the name of the company using Word art
d. Give details of the Company in brief
e. Use bullets if necessary

## PROCEDURE:

Step 1: Open a new word document using File $\rightarrow$ New option.
Step 2: Go to Page Layout $\rightarrow$ Page Borders, under the Page border tab choose the appropriate style and color, etc, and click OK.

Step 3: Go to Page Layout $\rightarrow$ Page Color, choose the appropriate color.
Step 4: Go to Insert $\rightarrow$ Clip Art, search for a relevant picture from the collections and insert it in to the page.

Step 5: Go to Insert $\rightarrow$ Word Art, choose the appropriate style from the list, type the company name and click OK.

Step 6: Type the company details in the document and do the following steps for various styles.
a. For bullets and numbering, select the appropriate style from the menu and apply to the paragraph.

```
:三`決-
```

b. For alignment, select the particular word or statement or paragraph in the document then press CTRL+L (left) or CTRL+R (right) or CTRL+E (center) or CTRL+J (justify) or select the relevant button from the formattingtoolbar.

c. For bold facing, select the particular word or statement or paragraph then press CTRL+B or select the on the formatting toolbar.
d. For italic style, select the particular word then press CTRL+I or select the on the formatting toolbar.
e. For underlining, select the particular word or statement or paragraph then press CTRL+U or select the on the formatting toolbar.

$$
\text { B } I \underline{\mathrm{U}}
$$

f. For spelling corrections, select the particular word, then right click the mouse or click spelling and grammar button on the standard toolbar.

ABC
g. For saving the document, press CTRL+S or click on save button on standard tool bar or select the save option from the Filemenu.


## RESULT:

Thus an advertisement for a company with the relevant specifications was prepared using MS-WORD and the output was verified.

## Ex.No: 1.c

## WORKING WITH TABLES

## Date:

## Aim

To create a table and use table formats and conversion operations.

## Procedure

1. Open the new document
2. Select Table menu bar, then choose insert table option.

3. Type the following contents within the table and use table formats

| S.No | RollNo | Name | M1 | M2 |
| :--- | :--- | :--- | :--- | :--- |
| 1 | 1001 | Devika | 88 | 84 |
| 2 | 1002 | Kavitha | 87 | 97 |
| 3 | 1003 | Bhuvana | 90 | 98 |
| 4 | 1004 | Prabu | 89 | 88 |
| 5 | 1005 | Rani | 79 | 93 |

4. Select Table Layout $\rightarrow$ Convert $\rightarrow$ Table to Text option to convert a table into text by separating commas and Select Comma in the Table To Text option.
S.No, RollNo, Name, M1, M2

1, 1001, Devika, 88, 84

2, 1002, Kavitha, 87, 97
3, 1003, Bhuvana, 90, 98
4, 1004, Prabu, 89, 88
5, 1005, Rani, 79, 93
5. Reverse the text into table by selecting Table $\rightarrow$ Convert $\rightarrow$ Text to Table option.

| S.No | RollNo | Name | M1 | M2 |
| :--- | :--- | :--- | :--- | :--- |
| 1 | 1001 | Devika | 88 | 84 |
| 2 | 1002 | Kavitha | 87 | 97 |
| 3 | 1003 | Bhuvana | 90 | 98 |
| 4 | 1004 | Prabu | 89 | 88 |
| 5 | 1005 | Rani | 79 | 93 |

## Result

Thus the table has been created and conversion operation is performed.

## Ex.No: 1.d

## MAIL MERGE AND LETTER PREPARATION

## Date:

Aim

To prepare letters and send it to multiple recipients by using Mail merge.

## Procedure

1. Open the new document and type the following letter

From
The Head of the Department
Department of CSE
E.G.S Pillay Engineering college

Nagapattinam

To

Our Students are interested to visit your company on the first week of Next month. I kindly request you to give permission to us.

Thanking You,
Yours faithfully,
HOD
2. Select Tools $\rightarrow$ Letters and Mailings $\rightarrow$ MailMerge Wizard
3. Select document type as Letters
4. Select starting document as Use the current document
5. Select recipients as Type a new list
6. Select Create...
7. New address list window appears
8. Type the details which you want
9. Save the address list as myaddress and goto next
10. Select Write your letter
11. Place the cursor at the To address in the document
12. Select the Address block.. and click OK
13. Select the Greetline.. and click OK
14. Now the document looks like..

From
The Head of the Department
Department of CSE
E.G.S Pillay Engineering college

Nagapattinam
To
««AddressBlock»"
««GreetingLine»»
Our Students are interested to visit your company on the first week of Next month. I kindly request you to give permission to us.

Thanking You,
Yours faithfully,
HOD
15. Select Preview your letters
16. Now the letter is ready for multiple recipients.
17. See all letters by clicking >> in the mail merge window.

## Output

From
The Head of the Department
Department of CSE
E.G.S Pillay Engineering college

Nagapattinam

To
Mr. Rajan
HCL Corporation
Chennai

Dear Mr.Rajan,

Our Students are interested to visit your company on the first week of Next month. I kindly request you to give permission to us.

Thanking You,

> Yours faithfully,

HOD

## From

The Head of the Department
Department of CSE
E.G.S Pillay Engineering college

Nagapattinam
To
Mr. Karthick
CTS Technologies
Chennai

Dear Mr.Karthick,

Our Students are interested to visit your company on the first week of Next month. I kindly request you to give permission to us.

Thanking You,

> Yours faithfully,

HOD

## Result

Thus the letter has been created for multiple recipients using mail merge.

Ex.No: 1.e
DRAWING - FLOWCHART

## Date:

Aim

To create a presentation in MS PowerPoint

## Procedure

1. Select Insert $\rightarrow$ shapes $\rightarrow$ Flowchart

2. Select the Shape and draw the flowchart


## Result

Thus the flowchart has been drawn using Auto shape tools

Ex.No: 1.f
SPREADSHEET-CHART

## Date:

Aim

To create a various types of chart in Excel.

## Procedure

1. Type the following

| RollNo | Name | Weight |
| :---: | :---: | :---: |
| 1001 | aaa | 63 |

1002 bbb 52

1003 CCC 46

1004 ddd 58

1005 eee 60
2. Select all data in the worksheet
3. Select Insert $\rightarrow$ Chart
4. Choose chart type and select Next and select Next
5. Type the chart title
6. Type X-axis title as Name and RollNo
7. Type Y-axis title as Weight
8. Click Next
9. Click Finish



## BAR CHART



PIE CHART

Weight chart


■ 1001 aaa - 1002 bbb - 1003 ccc ■ 1004 ddd

■ 1005 eee

## Result

Thus the charts - Line, XY, Bar and Pie were created.

## Ex.No: 1.g

SPREAD SHEET - FORMULA

## Date:

## Aim

To perform some calculation in spreadsheet by using Formula editor.

## Procedure

1. Type the following in Excel

RollNo Name English Maths Computer Total Average

| 1 | vimala | 63 | 57 | 62 |
| :--- | :--- | :--- | :--- | :--- |
| 2 | Priya | 52 | 56 | 54 |
| 3 | Divya | 96 | 89 | 90 |
| 4 | Sudha | 84 | 87 | 88 |
| 5 | Kavi | 75 | 76 | 74 |

2. Type the formula $=C 2+D 2+E 2$ in the cell $F 2$. Now the total appears in the cell
3. Copy the formula in the remaining total cells F3,F4,F5 and F6
4. Type the formula =F2/3 in the cell G2. Now the average appears in the cell
5. Copy the average formula in the remaining average cells G3,G4,G5 and G6

## Output

RollNo Name English Maths Computer Total Average

| 1 | vimala | 63 | 57 | 62 | 182 | 60.66 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 2 | Priya | 52 | 56 | 54 | 162 | 54 |


| 3 | Divya | 96 | 89 | 90 | 275 | 91.66 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 4 | Sudha | 84 | 87 | 88 | 259 | 86.33 |
| 5 | Kavi | 75 | 76 | 74 | 225 | 75 |

## Result

Thus the total and average were calculated using Formulas.

Ex.No: 1.h
SPREAD SHEET - EMPLOYEE SALARY SLIP

## Date:

## Aim

To create employee salary slip in MS Excel

## Procedure

1. Create table with fields below

Employee Code No = keep it same like in picture
Name = keep it same like in picture
Designation = keep it same like in picture
Basic Salary = keep it same like in picture

| EMPLOYEE <br> CODE NO | NAME | DESIGNATION | SALARY |
| :---: | :---: | :--- | :---: |
| 1001 | A | EXECUTIVE | 50000 |
| 1002 | B | OFFICER | 70000 |
| 1003 | C | ASSISTANT | 10000 |
| 1004 | D | EXECUTIVE | 65000 |
| 1005 | E | OFFICER | 35000 |
| 1006 | F | ASSISTANT | 14000 |
| 1007 | G | EXECUTIVE | 78000 |
| 1008 | H | OFFICER | 25000 |
| 1009 | I | ASSISTANT | 23000 |
| 1010 | J | EXECUTIVE | 15000 |
| 1011 | K | OFFICER | 25600 |
| 1012 | L | ASSISTANT | 14560 |
| 1013 | M | EXECUTIVE | 78000 |
| 1014 | N | OFFICER | 45000 |


| 1015 | O | ASSISTANT | 47800 |
| :---: | :---: | :--- | :---: |
| 1016 | P | EXECUTIVE | 14000 |
| 1017 | Q | OFFICER | 10000 |
| 1018 | R | ASSISTANT | 5000 |
| 1019 | S | EXECUTIVE | 25600 |
| 1020 | T | OFFICER | 25300 |

2. Now calculate some ALLOWANCES based on the following assumption

- House Rent Allowance (HRA) = If $10 \%$ of basic salary is higher than Rs 2000, than HRA will Rs 2000. If $10 \%$ of basic salary is lower than Rs 2000, then HRA will $10 \%$ of Basic Salary Formula $=\operatorname{IF}(D 2 * 10 \%>2000,2000, D 2 * 10 \%)$.
- Write formula in E2 cell \& drag formula to E21 cell,
- Dearness Allowance $(D A)=D A$ is $25 \%$ on Basic Salary Formula =D2*25\%
write formula in F2 cell \& drag formula to F21 cell
- Medical Allowance (MA) = Executives get MA Rs 1000 , Officers get MA Rs 700 \& Assistants get MA Rs 500
Formula $=I F(C 2=$ "Executive",1000,IF(C2="Officers",700,500))
Write formula in G2 cell \& drag formula to G21 cell

| EMPLOYEE <br> CODE NO | NAME | DESIGNATION | BASIC <br> SALARY | HRA(HOUSE <br> RENT <br> ALLOWANCE | DA(DEARNESS <br> ALLOWANCE) | MA(MEDICAL <br> ALLOWANCE) |
| :---: | :---: | :--- | :---: | :---: | :---: | :---: |
| 1001 | A | EXECUTIVE | 50000 | 2000 | 12500 | 1000 |
| 1002 | B | OFFICER | 70000 | 2000 | 17500 | 700 |
| 1003 | C | ASSISTANT | 10000 | 1000 | 2500 | 500 |
| 1004 | D | EXECUTIVE | 65000 | 2000 | 16250 | 1000 |
| 1005 | E | OFFICER | 35000 | 2000 | 8750 | 700 |
| 1006 | F | ASSISTANT | 14000 | 1400 | 3500 | 500 |
| 1007 | G | EXECUTIVE | 78000 | 2000 | 19500 | 1000 |


| 1008 | H | OFFICER | 25000 | 2000 | 6250 | 700 |
| :---: | :---: | :--- | :---: | :---: | :---: | :---: |
| 1009 | I | ASSISTANT | 23000 | 2000 | 5750 | 500 |
| 1010 | J | EXECUTIVE | 15000 | 1500 | 3750 | 1000 |
| 1011 | K | OFFICER | 25600 | 2000 | 6400 | 700 |
| 1012 | L | ASSISTANT | 14560 | 1456 | 3640 | 500 |
| 1013 | M | EXECUTIVE | 78000 | 2000 | 19500 | 1000 |
| 1014 | N | OFFICER | 45000 | 2000 | 11250 | 700 |
| 1015 | O | ASSISTANT | 47800 | 2000 | 11950 | 500 |
| 1016 | P | EXECUTIVE | 14000 | 1400 | 3500 | 1000 |
| 1017 | Q | OFFICER | 10000 | 1000 | 2500 | 700 |
| 1018 | R | ASSISTANT | 5000 | 500 | 1250 | 500 |
| 1019 | S | EXECUTIVE | 25600 | 2000 | 6400 | 1000 |
| 1020 | T | OFFICER | 25300 | 2000 | 6325 | 700 |

3. Now calculate Gross salary, professional tax, salary paid per month and annual salary/net salary based on the following assumption

- Gross Salary = Total of Basic + HRA + DA + MA Formula =SUM(D2:G2)
write formula in H 2 cell \& drag formula to H 21 cell
- Professional Tax = Upto $5000=0$, upto $1000=60$, upto $15000=100 \&$ over $15000=150$

Formula $=\operatorname{IF}(H 2<=5000,0, \operatorname{IF}(H 2<=10000,60, \operatorname{IF}(H 2<=15000,110,150))$ )
Write formula in 12 cell \& drag formula to 121 cell

- Salary Paid Per Month = Gross Salary - Professional Tax Formula $=\mathrm{H} 2-\mathrm{I} 2$
Write formula in J2 cell \& drag formula to J21 cell
- Annual Salary/Net Salary = Salary Paid Per Month * 12

Formula $=J 2 * 12$
Write formula in K2 cell \& drag formula to K21 cell

## Output: Salary slips of Employees

| $\begin{gathered} \text { EMPLO } \\ \text { YEE } \\ \text { CODE } \\ \text { NO } \end{gathered}$ | $\begin{aligned} & \text { NA } \\ & \text { ME } \end{aligned}$ | DESIGNATIO N | BASIC <br> SALAR <br> Y | HRA(H <br> OUSE <br> RENT <br> ALLOW <br> ANCE | DA(DEARN ESS ALLOWAN CE) | MA(MEDI CAL ALLOWAN CE) | GROSS <br> SALARY | PROFES SIONAL TAX | SALARY <br> PAID <br> PER <br> MONTH | ANNUAL SALARY/ NET SALARY |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1001 | A | EXECUTIVE | 50000 | 2000 | 12500 | 1000 | 65500 | 150 | 65350 | 784200 |
| 1002 | B | OFFICER | 70000 | 2000 | 17500 | 700 | 90200 | 150 | 90050 | 1080600 |
| 1003 | C | ASSISTANT | 10000 | 1000 | 2500 | 500 | 14000 | 110 | $13890$ | 166680 |
| 1004 | D | EXECUTIVE | 65000 | 2000 | 16250 | 1000 | 84250 | 150 | 84100 | 1009200 |
| 1005 | E | OFFICER | 35000 | 2000 | 8750 | 700 | 46450 | 150 | 46300 | 555600 |
| 1006 | F | ASSISTANT | 14000 | 1400 | 3500 | 500 | 19400 | 150 | 19250 | 231000 |
| 1007 | G | EXECUTIVE | 78000 | 2000 | 19500 | 1000 | 100500 | 150 | 100350 | 1204200 |
| 1008 | H | OFFICER | 25000 | 2000 | 6250 | 700 | 33950 | 150 | 33800 | 405600 |
| 1009 | 1 | ASSISTANT | 23000 | 2000 | 5750 | 500 | 31250 | 150 | 31100 | 373200 |
| 1010 | J | EXECUTIVE | 15000 | 1500 | 3750 | 1000 | 21250 | 150 | 21100 | 253200 |
| 1011 | K | OFFICER | 25600 | 2000 | 6400 | 700 | 34700 | 150 | 34550 | 414600 |
| 1012 | L | ASSISTANT | 14560 | 1456 | 3640 | 500 | 20156 | 150 | 20006 | 240072 |
| 1013 | M | EXECUTIVE | 78000 | 2000 | 19500 | 1000 | 100500 | 150 | 100350 | 1204200 |
| 1014 | N | OFFICER | 45000 | 2000 | 11250 | 700 | 58950 | 150 | 58800 | 705600 |
| 1015 | 0 | ASSISTANT | 47800 | 2000 | 11950 | 500 | 62250 | 150 | 62100 | 745200 |
| 1016 | P | EXECUTIVE | 14000 | 1400 | 3500 | 1000 | 19900 | 150 | 19750 | 237000 |
| 1017 | Q | OFFICER | 10000 | 1000 | 2500 | 700 | 14200 | 110 | 14090 | 169080 |
| 1018 | R | ASSISTANT | 5000 | 500 | 1250 | 500 | 7250 | 60 | 7190 | 86280 |
| 1019 | S | EXECUTIVE | 25600 | 2000 | 6400 | 1000 | 35000 | 150 | 34850 | 418200 |
| 1020 | T | OFFICER | 25300 | 2000 | 6325 | 700 | 34325 | 150 | 34175 | 410100 |

## Result:

Thus salary slip for employees is created using MS Excel

Ex.No: 1.i

## CREATING A POWERPOINT PRESENTATION

## Date:

Aim

To create a presentation in MS PowerPoint

## Procedure

1. When PowerPoint starts, you will see a window like this one.

2. Move your mouse to the rectangle that says Click to add title, click once, then start typing
3. Move your mouse to rectangle that says Click to add subtitle, click once, then start typing
4. Move your mouse to right end of the toolbar at the top of your PowerPoint Window and click New Slide, once ... a new slide will appear
5. Move your mouse to the Insert Menu and click once to activate the menu
6. Choose Picture and Choose Clip Art
7. To change slide design :
a. Move your mouse to the Format Menu and click once to activate the menu
b. Choose Slide Design ...
c. The Slide Design Tool will now be open on the right side of your PowerPoint window.
d. Scroll down to see the different designs available for use.
e. Once you find the design you want to use, click on it once and it will become the slide design on all your slides.
f. The Slide Design Tool will now be open on the right side of your PowerPoint window.
g. Scroll down to see the different designs available for use.
h. Once you find the design you want to use, click on it once and it will become the slide design on all your slides.
8. To add animation:
a. Move your mouse to the Slide Show Menu and click once to activate the menu
b. Choose Custom Animation ...
c. Click once on the object you want to animate to activate it
d. Move your mouse to the Custom Animation tool that appears on the right side of the window and click on Add Effect
e. Choose the type of effect you want in the submenu that appears
f. Choose the specific effect in the next submenu
g. Add animation to all the objects you want animated using the steps outlined.
h. When you are done you will see numbers next to the animated objects
i. You can put them in a different order by clicking and dragging them in their list
j. You can change how the animation occurs by clicking on the "down arrow" at the end of each effect $\rightarrow$ a new submenu will appear from which you can make more choices.

## Result

Thus a presentation is created using MS PowerPoint

## Ex.No:2.a

## PROGRAM TO PRINT TEXT

## Date:

## AIM

To write a c program to print the given text

## ALGORITHM

1. Start the program
2. Display the given text using printf(); function
3. Stop the program

## PROGRAM

\# include <stdio.h> \# include <conio.h> void main()
\{
clrscr();
printf("E.G.S Pillay Engineering College"); printf("\nNagapattinam"); getch();
\}

## OUTPUT

E.G.S Pillay Engineering College

Nagapattinam

## Result

Thus the program to print given text is executed successfully and its inputs and outputs are verified

## Ex.No: 2.b PROGRAM TO READ TWO NUMBERS AND PRINT THE SUM OF GIVEN TWO NUMBERS

Date:

## AIM

To write a c program to read two numbers and print their sum

## ALGORITHM

1. Declare required variables
2. Get two numbers as input
3. Calculate the sum of two numbers and display it

## PROGRAM

\# include <stdio.h> \# include <conio.h> void main()
\{

```
int a,b, sum;
clrscr ();
printf ("ENTER VALUE FOR A : ");
scanf ("%d",&a);
printf("ENTER VALUE FOR B :");
scanf("%d",&b);
sum=a+b;
printf("SUM OF GIVEN TWO NUMBERS IS : %d", sum);
getch();
```

\}

## OUTPUT

ENTER VALUE FOR A :10
ENTER VALUE FOR B : 5
SUM OF GIVEN TWO NUMBERS IS : 15

## Result

Thus the program to two numbers and print the sum of given two numbers is executed successfully and its inputs and outputs are verified

## Ex.No: 2.c PROGRAM TO FIND BIGGEST OF TWO NO BY USING CONDITIONAL/TERNARY OPERATOR

## Date:

## AIM

To write a c program to biggest of two no by using Conditional/Ternary operator

## ALGORITHM

1. Declare required variables
2. Get two numbers as input
3. Find the biggest of two numbers and display it

## PROGRAM

\#include <stdio.h>
\#include <conio.h> void main( )
\{
int a,b,big;
clrscr( );
printf("enter the value a :");
scanf("\%d",\&a);
printf("enter the value of $b$ :");

```
scanf("%d",&b);
big=(a>b)?a:b;
printf("biggest of the given numbers is : %d",big);
getch();
}
```


## OUTPUT

enter the value a : 10
enter the value b: 5
biggest of the given numbers is :10

## Result

Thus the program to find biggest of two no by using conditional/ternary operator is executed successfully and its inputs and outputs are verified.

## Ex.No: 2.d

PROGRAM TO CALCULATE SIMPLE INTEREST

## Date:

## AIM

To write a C program to implement simple interest

## ALGORITHM

1. Start
2. Define variables: amount, rate, time, si;
3. Take input from keyboard for all the input variables
4. Calculate the simple interest and compound interest
5. Display the simple interest value
6. Stop.

## PROGRAM

\#include<stdio.h>
\#include<conio.h> void main()

int amount, rate, time, si;
clrscr():
printf("\nEnter Principal Amount : ");
scanf("\%d", \&amount);
printf("\nEnter Rate of Interest : ");
scanf("\%d", \&rate);
printf("\nEnter Period of Time : ");
scanf("\%d", \&time);
si = (amount * rate * time) / 100;
printf("\nSimple Interest : \%d", si);
\}

## OUTPUT

Enter Principal Amount: 500
Enter Rate of interest: 5
Enter Period of Time : 2
Simple Interest: 50

## Result

Thus the program to calculate simple interest is successfully executed and its inputs and outputs are verified.

## Ex.No: 3.a PROGRAM TO READ THREE NUMBERS AND PRINT THE BIGGEST OF GIVEN THREE NUMBERS Date:

AIM:
To write a c program to examine the biggest of given three numbers.

## ALGORITHM:

1. Declare three integer variables
2. Read the 3 inputs
3. Compare ((a>b)\&\&(a>c)). If true then display a as big
4. Else compare ( $b>c$ ). if true then display $b$ as big
5. Else display c as big.

## PROGRAM

```
# include <stdio.h>
# include <conio.h>
void main( )
int a,b,c;
clrscr( );
printf("ENTER VALUE FOR A:");
scanf("%d",&a);
printf("ENTER VALUE FOR B:");
scanf("%d",&b);
print("ENTER VALUE FOR C:");
scanf("%d",&c);
if((a>b)&&(a>c))
{
```

printf ("BIGGEST OF ABOVE GIVEN THREE NUMBER IS \%d",a);
\}
else if(b>c)
\{
printf ("BIGGEST OF ABOVE GIVEN THREE NUMBER IS \%d",b);
\}
else
\{
printf ("BIGGEST OF ABOVE GIVEN THREE NUMBER IS \%d",c);
\}
getch( );
\}

## OUTPUT

ENTER VALUE FOR A : 8
ENTER VALUE FOR A : 2
ENTER VALUE FOR A : 5
bIGGeST OF ABOVE GIVEN THREE NUMBER IS 8

## Result:

Thus the program to read three numbers and print the biggest of given three numbers is successfully executed and its inputs and outputs are verified.

## Ex.No: 3.b PROGRAM TO READ A NUMBER AND FIND WHETHER THE GIVEN NUMBER IS EVEN OR ODD

## Date:

## AIM:

To write a c program to check whether given Number is odd or even.

## ALGORITHM:

1. Declare a variable to get a Number
2. Read the input
3. Get the remainder of given number using modulo operator
4. If remainder is 0 prints "Even Number", else print "Odd Number".

## PROGRAM

```
# include <stdio.h>
# include <conio.h>
void main()
{
    int n,r;
    clrscr();
    printf("ENTER A NUMBER :");
    scanf("%d", &n);
    r=n%2;
    if(r= = 0)
        printf("The above given number is even number");
    else
        printf("The above given number is odd number");
    getch();
}
```


## OUTPUT

ENTER A NUMBER : 2
The above given number is even number

## Result:

Thus the program to read a number and to find whether the given number is even or odd is successfully executed and its inputs and outputs are verified.

## Ex.No: 3.C PROGRAM TO ACCEPT A YEAR AND CHECK WHETHER THE GIVEN YEAR IS LEAP YEAR OR NOT

## Date:

## AIM:

To write a c program to check whether the given year is leap year or not.

## ALGORITHM:

1. Read the year as input
2. Extract the remainder from division operation of year by 4
3. If remainder is 0 print "Given year is Leap year" else print "Given year is not a Leap year"

## PROGRAM

```
    # include <stdio.h>
    # include <conio.h>
    main( )
    {
        int y;
        clrscr( );
        printf("Enter a year:");
        scanf("%d",&y);
        if((y%4==0)&&(y%100!=0)|| (y%400==0))
                printf("The above given year is a leap year");
    else
                printf("The above given year is not a leap year");
        getch();
```

    \}
    
## OUTPUT

Enter a year : 2012
The above given year is not a leap year

## Result:

Thus the program to accept a year and to check whether the given year is leap year or not is successfully executed and its inputs and outputs are verified.

Ex.No: 3.d

## Date:

## AIM:

To write a menu driven c program to implement an Arithmetic Calculator.

## ALGORITHM:

1. Get the two numbers
2. Enter the choice
3. Pass the choice into switch case
4. In case 1 , add the two numbers and print the result
5. In case 2 , subtract the two numbers and print the result
6. In case 3 , multiply the two numbers and print the result Step
7. In case 4 , divide the two numbers and print the result
8. In case 5 , exit the program

## PROGRAM

```
\#include<stdio.h>
\#include<conio.h>
\#include<process.h>
void main()
\{
    int n1, n2, ch ;
    clrscr() ;
    printf("Enter the first number:");
    scanf("\%d", \&n1);
    printf("\nEnter the second number:");
    scanf("\%d", \&n2);
    printf(" \(\mathrm{n}[1]\)-> Addition ") ;
    printf("\n[2] -> Subtraction ");
    printf("\n[3] -> Multiplication ");
    printf("\n[4] -> Division") ;
    printf("\n[5] -> Exit ") ;
    printf("\n\nEnter your choice <1...4> : ");
    scanf("\%d", \&ch);
    switch(ch)
    \{
    case 1 :
        printf("\n\%d + \%d = \%d", n1, n2, n1 + n2) ;
        break;
    case 2 :
        printf("\n\%d - \%d = \%d", n1, n2, n1-n2) ;
        break;
    case 3 :
        printf("\n\%d * \%d = \%d", n1, n2, n1 * n2);
        break;
    case 4 :
        printf("\n\%d / \%d = \%.2f", n1, n2, (float)n1 / n2);
        break;
    case 5 :
        exit(0);
    default:
        printf("\nInvalid choice");
```

```
            break;
    }
    getch();
    }
```


## OUTPUT

Enter the first number : 15
Enter the second number : 3
[1] $->$ Addition
[2] -> Subtraction
[3] -> Multiplication
[4] -> Division
[5] -> Exit
Enter your choice <1...5> : 4
$15 / 3=5.00$

## Result:

Thus the program to perform arithmetic operations using switch-case is successfully executed and its inputs and outputs are verified.

## Ex.No: 3.e PROGRAM TO ACCEPT STUDENT ROLL NO, MARKS IN 3 SUBJECTS AND CALCULATE

TOTAL, AVERAGE, GRADE AND PRINT IT.

## Date:

## AIM

To write a c program to calculate students mark details

## ALGORITHM

1. Declare required variables
2. Get students roll no, marks as input
3. Calculate total and average, grade and display it

## PROGRAM

```
# include <stdio.h>
# include <conio.h>
void main()
    int r,b,c,d, tot, avg;
    char grade[30];
    clrscr();
    printf ("ENTER STUDENT RNO :");
scanf ("%d",&r);
printf("ENTER FIRST SUBJECT MARKS :");
scanf("%d",&b);
printf("ENTER SECOND SUBJECT MARKS:");
scanf("%d",&c);
printf("ENTER THIRD SUBJECT MARKS :");
```

```
scanf("%d",&d);
tot=b+c+d;
avg=tot/3;
printf("\n\n\t\t E.G.S PILLAY ENGINEERING COLLEGE \n\n");
printf("\n STUDENT RNO : %d ",r);
printf("\n FIRST SUBJECT MARKS : %d ",b);
printf("\nSECOND SUBJECT MARKS : %d ",C);
printf("\nTHIRD SUBJECT MARKS : %d ",d);
printf("\nTOTAL MARKS: %d",tot);
printf("\nAVERAGE MARKS : %d", avg);
if(avg>=60)
{
    printf("First Class");
}
else if(avg>=50)
{
    printf("Second Class");
}
else if(avg>=35)
{
    printf("Third Class");
}
else
{
    printf("Fail");
}
getch();
```

\}

## OUTPUT

ENTER STUDENT RNO :1001
ENTER FIRST SUBJECT MARKS : 80
ENTER SECOND SUBJECT MARKS:90
ENTER THIRD SUBJECT MARKS :80
E.G.S PILLAY ENGINEERING COLLEGE

STUDENT RNO:1001
FIRST SUBJECT MARKS :80
SECOND SUBJECT MARKS:90
THIRD SUBJECT MARKS :80
TOTAL MARKS:250
AVERAGE MARKS :83
First Class

## Result

Thus the program to accept student roll no, marks in 3 subjects and calculate total, average and grade is executed successfully and its inputs and outputs are verified.

## Ex.No: 4.a

## PROGRAM TO PRINT ODD NUMBERS FROM 1 TO 10

## Date:

## AIM:

To write a c program to print odd numbers from 1 to 10

## ALGORITHM:

1. Initialize for loop from 1 to 10
2. Get the odd numbers by incrementing the loop by 2
3. Display the odd numbers.

## PROGRAM

\# include <stdio.h> \# include <conio.h> void main()
\{
int i ;
clrscr();
for ( $\mathrm{i}=1 ; \mathrm{i}<=10$; $\mathrm{i}+=2$ )
\{
printf("\%d\n",i);
\}
getch( );
\}

## OUTPUT

1
3
5
7
9

## Result

Thus the program to print odd numbers from 1 to 10 is executed successfully and its inputs and outputs are verified.

Ex.No: 4.b

## Date:

## AIM:

To write a c program to find the sum of 10 natural numbers.

## ALGORITHM:

1. Initialize the sum as 0
2. Initialize a counter with 1
3. Overwrite the sum by adding counter value $\&$ sum
4. Increment the counter value by 1
5. Repeat the steps $4 \& 5$ until the counter is less than or equal to 10
6. Print the sum

## PROGRAM

```
#include <stdio.h>
#include <conio.h>
void main( )
    {
        int n,sum=0,i;
        clrscr( );
        for (i=1; i<=10;i++)
        {
            sum=sum+i;
    }
    printf("Sum of natural numbers from 1 to 10 is %d\n",sum);
    getch( );
}
```


## OUTPUT

Sum of natural numbers from 1 to 10 is 55

## Result

Thus the program to print sum of the natural numbers from 1 to 10 is executed successfully and its inputs and outputs are verified.

Ex.No: 4.c
PROGRAM TO ACCEPT A NUMBER AND PRINT MATHEMATICAL TABLE

## OF THE GIVEN NO

## Date:

## AIM:

To write a c program to display given multiplication table

## ALGORITHM:

1. Get the table no
2. Initialize for loop with 1
3. Perform multiplication for first five numbers
4. Print the sum at each iteration

## PROGRAM

\#include<stdio.h>
\#include<conio.h>
void main()
\{
int i,tab,n,res;
printf("\nEnter the multiplication table:");
scanf("\%d",\&tab);
printf("\nEnter the limit of multiplication table:");
scanf("\%d",\&n);
for $(i=1 ; i<=n ; i++)$
\{
res=i*tab;
printf("<br>%d * \%d = \%d $\left.{ }^{\prime} n^{\prime \prime}, i, t a b, r e s\right) ;$
\}
getch();
\}

## OUTPUT

Enter the multiplication table:2
Enter the limit of multiplication table:5
$1 * 2=2$
$2 * 2=4$
$3 * 2=6$
4*2=8
$5 * 2=10$

## Result

Thus the program to a number and print mathematical table of the given no is executed successfully and its inputs and outputs are verified.

Ex.No: 4.d

## PROGRAM TO PRINT FIBONACCI SERIES

## Date:

## AIM:

To write a c program to display Fibonacci series

## ALGORITHM:

1. Get the limit for displaying the series
2. Initially print the value of $a$ and $b$
3. Print the remaining series by adding previous value
4. Display the fibonacci series

## PROGRAM

\#include<stdio.h>
\#include<conio.h>
void main()
\{
int $a, b, c, r ;$
clrscr();
printf("Enter the range : ");
scanf("\%d",\&r);
$a=0, b=1$;
printf("\%d", a);
printf("\n\%d", b);
printf(" $\backslash \mathrm{n}$ ");
c=0;
do
\{
$c=a+b ;$
if(c<=r)
printf("\%d\n", c);
$a=b$;
$b=c ;$
\}while(c<=r);
getch();
\}

## OUTPUT

Enter the limit : 10
0
1

1
2
3
5
8

## Result

Thus the program to print fibonacci series is executed successfully and its inputs and outputs are verified.

## Ex.No: 4.e

## PROGRAM TO CHECK ARMSTRONG NUMBER

## Date:

## AIM:

To write a c program to check whether the given no is Armstrong or not.

## ALGORITHM:

1. Copy the given number into a variable b. Initialize a counter to 1 and sum to 0
2. Extract the remainder of given number while dividing 10
3. Calculate the value of remainder by assigning power 3
4. Overwrite the sum by adding above result with available sum
5. Overwrite the number by divide with 10
6. Repeat the steps 1 to 5 until the number is greater than 0
7. Compare the sum and copy of the number
8. If they are equal print as "Armstrong" else print "Not Armstrong"

## PROGRAM

```
#include <stdio.h>
#include <conio.h>
    void main()
{
    int n, n1, rem, num=0;
    clrscr();
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    n1=n;
```

    while(n1!=0)
    \{
    ```
            rem=n1%10;
            num+=rem*rem*rem; n1/=10;
        }
if(num==n)
{
    printf("\n%d is an Armstrong number",n);
}
else
{
    printf("\n%d is not an Armstrong number",n);
}
```


## OUTPUT

Enter a positive integer: 371
371 is an Armstrong number.

## Result

Thus the program to check armstrong number is executed successfully and its inputs and outputs are verified.

## Ex.No: 4.f

PROGRAM TO REVERSE THE GIVEN NUMBER AND TO CHECK PALINDROME

## Date:

## AIM:

To write a c program to reverse the given number and to check palindrome.

## ALGORITHM:

1. Get the number to reverse
2. Reverse the given number using the logic given in the program
3. Check for palindrome
4. Print the result of palindrome

## PROGRAM:

\#include<stdio.h>
\#include<conio.h> void main()
\{

```
int n,rev=0;
clrscr();
printf("Enter a number to reverse :");
scanf("%d",&n);
while(n!=0)
{
    rev=rev*10;
    rev=rev+(n%10);
    n=n/10;
}
```

```
printf("Reverse of given no is :%d\n",rev);
if(n==rev)
{
    printf("Palindrome");
}
else
{
    printf("Not Palindrome");
}
getch():
}
```


## Result

Thus the program to reverse the given number and to check palindrome is executed successfully and its inputs and outputs are verified.

## Ex.No: 4.g

## PROGRAM TO UNDERSTAND NESTED FOR LOOP

## Date:

## AIM:

To write a c program to understand nested for loop.

## ALGORITHM:

5. Get the limit for for loop
6. The range for outer for loop is 1 to $n$
7. The range for inner for loop is 1 to $i$
8. Print the character * from inner for loop

## PROGRAM

\#include <stdio.h> \#include <conio.h> void main()

int i,j,n; clrscr();
printf("Enter the limit : ");
scanf("\%d",\&n);
for(i=1;i<=n;i++)
\{

```
for(j=1;j<=i;j++)
{
printf("*");
```

```
            }
                printf("\n);
            }
        getch();
    }
```


## OUTPUT

Enter the limit : 5
*
**
***
****

## Result

Thus the program to understand nested for loop is executed successfully and its inputs and outputs are verified

## Ex.No: 5.a

SUM OF ARRAY ELEMENTS
Date:

## AIM:

To write a c program to find the sum of given array elements.

## ALGORITHM:

Step 1: Declare an array with necessary size
Step 2: Get the value for total number of elements
Step 3: Initialize an index value to 0
Step 4: Read the input
Step 5: Increment the index value by 1
Step 6: Repeat steps $4 \& 5$ until counter less than total no. of elements
Step 7: Initialize an index value to 0 and sum to 0
Step 8: Obtain the sum by adding current index array value with available
Sum
Step 9: Increment the index value by 1
Step 10: Repeat steps $8 \& 9$ until index value less than total no. of elements
Step 11: Print the sum

## PROGRAM:

```
#include<stdio.h>
void main()
{
    int i,n,a[10],sum=0;
    printf("Enter total no. of Elements\n");
    scanf("%d",&n);
    printf("Enter Array elements one by one\n");
```

```
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
    {
            sum=sum+a[i];
}
printf("The Sum of Array Elements is %d\n",sum);
getch();
}
```


## OUTPUT:

```
Enter total no. of Elements
8
Enter Array elements one by one
15
69
32
10
45
66
32
11
The Sum of Array Elements is 280
```


## Result

Thus the program to find the sum of given array elements is executed successfully and its inputs and outputs are verified.

Ex.No: 5.b
DISPLAY EVEN NUMBERS OF AN ARRAY
Date:

## AIM:

To write a c program to print the even numbers of given array elements.

## ALGORITHM:

Step 1: Declare an array with necessary size
Step 2: Get the value for total number of elements
Step 3: Initialize an index value to 0
Step 4: Read the input
Step 5: Increment the index value by 1
Step 6: Repeat steps $4 \& 5$ until counter less than total no. of elements
Step 7: Initialize an index value to 0

Step 8: Extract the remainder by dividing array index value with 2
Step 9: If the remainder is 0 print the value
Step 10: Increment the index value by 1
Step 11: Repeat steps 8 to 10 until index value less than total no. of elements

## PROGRAM:

```
#include<stdio.h>
void main()
{
        int i,n,a[20];
        printf("Enter total no. of Elements\n");
        scanf("%d",&n);
        printf("Enter Array elements one by one\n");
        for(i=0;i<n;i++)
        {
            scanf("%d",&a[i]);
        }
        printf("The even numbers of given array:\n");
        for(i=0;i<n;i++)
    {
        if(a[i]%2==0)
        printf("%d\n",a[i]);
    }
    getch();
}
```


## OUTPUT:

Enter total no. of Elements
6
Enter Array elements one by one
98
11
35
61
22
14
The even numbers of given array:
98
22
14

## Result

Thus the program to print the even numbers of given array elements is executed successfully and its inputs and outputs are verified.

Ex.No: 5.c

## ADDITION OF MATRIXES

## Date:

## AIM:

To write a c program to perform matrix addition.

## ALGORITHM:

## Step 1: Start

Step 2: Declare the three dimensional integer arrays $a[3][3], b[3][3]$ and $c[3][3]$ and declare the variables $i$ and $j$ as integers.
Step 3: Read the input for the matrixes A and B.
Step 4: Print the matrixes $A$ and $B$.
Step 5: Add the matrixes $A$ and $B$ and print the result in a matrix $C$.
Step 6: Stop

## PROGRAM:

\#include<stdio.h>
\#include<conio.h>
void main()
\{

```
int a[3][3],b[3][3],c[3][3],i,j;
clrscr();
printf("Enter values of Matrix A : \n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
printf("Enter values of Matrix B : \n");
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&b[i][j]);
    }
}
```

```
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
        }
    }
    printf("Added Matrix\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
    getch();
}
```


## OUTPUT:

Enter values of Martix A:
3
3
3
Enter values of Matrix B:
3
3
3
Added Matrix
666
666
666

## Result

Thus the program to perform matrix addition is executed successfully and its inputs and outputs are verified.

Ex.No: 5.d

## MULTIPLICATION OF MATRIXES

## Date:

## AIM:

To write a c program to perform matrix multiplication.

## ALGORITHM:

## Step 1: Start

Step 2: Declare the three dimensional integer arrays $a[3][3], b[3][3]$ and $c[3][3]$ and declare the variables $\mathrm{k}, \mathrm{I}$ and j as integers.
Step 3: Read the input for the matrixes A and B.
Step 4: Print the matrixes A and B.
Step 5: Multiply the matrixes $A$ and $B$ and print the result in a matrix $C$. Step 6: Stop

## PROGRAM:

\#include<stdio.h>
\#include<conio.h> void main()
\{

```
    int a[3][3],b[3][3],c[3][3],i,j,k;
    clrscr();
    printf("Enter values of Matrix A :\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
    }
}
printf("Enter values of Matrix B : \n");
for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&b[i][j]);
        }
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
                c[i][j]=0;
```

```
                for(k=0;k<3;k++)
                {
                        c[i][j]=c[i][j]+a[i][k]*b[k][j];
                }
            }
        }
        printf("Multiplied Matrix\n");
        for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
    getch();
}
```


## OUTPUT:

Enter values of Martix A:
3
3
3
Enter values of Matrix B:
3
3
3
Multiplied Matrix
999
999
999

## Result

Thus the program to perform matrix multiplication is executed successfully and its inputs and outputs are verified.

## Ex.No: 6.a

## STRING PALINDROME CHECKING

## Date:

## AIM:

To write a c program to check whether the given string is palindrome or not

## ALGORITHM:

Step 1: Create a character array with necessary size
Step 2: Read the String
Step 3: Copy the String into another character array
Step 4: Get reverse string of input by using strrev function
Step 5: Compare the above result with copied string
Step 6: If two string s are same print "Palindrome" else print "Not
Palindrome"

PROGRAM:
\#include<stdio.h>
\#include<string.h>
main()
\{
char s[20],s1[20];
printf("Enter a String : \n");
scanf("\%s",s);
strcpy(s1,s);
if(strcmp(s,s1)==0)
printf("The Given String is Palindrome\n");
else
printf("The Given String is Not Palindrome\n");
\}

OUTPUT: Enter a String: noornilo
The Given String is not Palindrome

## Result

Thus the program to check whether the given string is palindrome or not is executed successfully and its inputs and outputs are verified.

Ex.No: 6.b

## STRING CONCATENATION

Date:

## AIM:

To write a c program to find the length of given two strings and concatenate them

## ALGORITHM:

Step 1: Create two character arrays with necessary size
Step 2: Read the Strings
Step 3: Calculate the string lengths using strlen function
Step 4: Print the string lengths
Step 5: Join the two strings using strcat function
Step 6: Print the concatenated string

## PROGRAM:

\#include<stdio.h>
\#include<string.h>
main()
\{
char s[20],s1[20];
printf("Enter a String1\n");
scanf("\%s",s);
printf("Enter a String2\n");
scanf("\%s",s1);
printf("Length of the String $s$ is \%d $\mathrm{n}^{\prime \prime}$,(strlen(s));
strcat(s,s1);
printf("The Concatenated String is : \%s $\backslash \mathrm{n} ", \mathrm{~s})$;
\}

## OUTPUT:

Enter a String1 : noor
Enter a String2 : nilo
Length of the String $s$ is 4
The Concatenated String is : noornilo

## Result

Thus the program to find the length of given two strings and concatenate them is executed successfully and its inputs and outputs are verified.

Ex.No: 6.c

## STRING COPY

## AIM:

To write a c program to copy the given string

## ALGORITHM:

Step 1: Create two character arrays with necessary size
Step 2: Read the Strings

Step 3: copy the string using strcpy function
Step 4: Print the copied string

## PROGRAM:

```
#include<stdio.h>
#include<string.h>
main()
{
char s[20],s1[20];
printf("Enter a String1 : \n");
scanf("%s",s);
printf("Enter a String2: \n");
scanf("%s",s1);
strcpy(s,s1);
printf("Copied String s is:%s",s);
}
```


## OUTPUT:

Enter a String1: noor
Enter a String2 : nilo
Copied String s is: nilo

## Result

Thus the program to copy the given string is executed successfully and its inputs and outputs are verified.

## Ex.No: 6.d

## STRING COMPARISION

## AIM:

To write a c program to check whether the given string is identical or not

## ALGORITHM:

Step 1: Create two character arrays with necessary size

## Step 2: Read the Strings

Step 3: Compare the string using strcmp function
Step 4: If the value returned is equal to zero, then print that two strings are identical
Step 5: Else print that the strings are not identical

## PROGRAM:

\#includesstdio.h>
\#include<string.h>
main()
\{
char s[20],s1[20];

```
int n;
printf("Enter a String1 : \n");
scanf("%s",s);
printf("Enter a String2 : \n");
scanf("%s",s1);
n=strcmp(s,s1);
if(n==0)
{
printf("Two Strings are identical");
}
Else
{
printf("Two Strings are not identical");
}
}
```


## OUTPUT:

Enter a String1 : nilo
Enter a String2 : nilo
Two Strings are identical

## Result

Thus the program to check whether the given string is identical or not is executed successfully and its inputs and outputs are verified.

## Ex.No: 6.e

## STRING REVERSE

## AIM:

To write a c program to reverse the given string

## ALGORITHM:

Step 1: Create a character arrays with necessary size
Step 2: Read the String
Step 3: Reverse the String using strrev function
Step 4: Print the reversed string

## PROGRAM:

\#include<stdio.h>
\#include<string.h>
main()
\{
char s[20];
printf("Enter a String : ");
scanf("\%s",s);
printf("Reversed String s is: \%s",(strrev(s));
\}

## OUTPUT:

Enter a String : noornilo
Reversed String is : olinroon

## Result

Thus the program to reverse the given string is executed successfully and its inputs and outputs are verified.

Ex.No: 7.a
FUNCTIONS WITHOUT ARGUMENTS \& RETURN TYPE

## Date:

## AIM:

To write a c program to check whether the given year is leap or not using functions.

## ALGORITHM:

Step 1: Create a function isleap()
Step 2: Inside the function
a. Read the year as input
b. Extract the remainder from division operation of year by 4
c. If remainder is 0 print "Given year is Leap year" else print "Given year is not a Leap year"
Step 3: Inside the main function call the isleap() function

## PROGRAM:

\#include<stdio.h> void isleap() \{
int yr;
printf("Enter a Year\n");
scanf("\%d",\&yr);
if( $\mathrm{yr} \% 4==0$ )
printf("Given Year is Leap year");
else
printf("Given Year is Not a Leap year");
\}
void main()
\{
clrscr()
isleap();

```
getch();
```

\}

OUTPUT:
Enter a Year
1965
Given Year is Not a Leap year

## Result

Thus the program to check whether the given year is leap or not using functions is executed successfully and its inputs and outputs are verified.

## Ex.No: 7.b FUNCTIONS WITHOUT ARGUMENTS \& WITH RETURN TYPE

Date:

## AIM:

To write a c program to calculate the area of triangle using functions.

## ALGORITHM:

Step 1: Create a function area()
Step 2: Inside the function
a. Read the 3 sides of triangle b. Calculate the sum of 3 sides
c. Divide the sum by 2 and store it into $s$
d. Subtract the sides from sand store them into variables e. Multiply s with above 3
results
f. Take the square root of above result g. Return the above
result as area
Step 3: Inside the main function call the function area()
Step 4: Print the area by obtaining the return value of area()

PROGRAM:
\#include<stdio.h>
\#include<math.h>
float area()
\{
int a,b,c;
float s,ar;
printf("Enter 3 Sides\n"); scanf("\%d\%d\%d",\&a,\&b,\&c);
$s=(a+b+c) / 2$;
ar=sqrt(s*(s-a)*(s-b)*(s-c));
return ar;
\}

```
float main()
{
float a;
a=area();
printf("The Area of Triangle is %f\n",a);
}
```


## OUTPUT:

Enter 3 Sides
12
8
7
The Area of Triangle is 19.748418

## Result

Thus the program to calculate the area of triangle using functions is executed successfully and its inputs and outputs are verified.

## Ex.No: 7.c FUNCTIONS WITH ARGUMENTS \& WITHOUT RETURN TYPE

Date:

## AIM:

To write a c program to sort the given array of elements using functions.

## ALGORITHM:

Step 1: Create a function sort()
Step 2: Inside the function
a. Initialize a index to 0
b. Initialize the sub index to counter +1
c. Compare the two numbers which are available in array index value and array sub index value
d. If the first number is greater than second number swap them e. Increment the sub index by 1
f. Repeat the steps c to e until sub index less than total number of elements
g. Increment the index by 1
h. Repeat the steps $b$ to $g$ until sub index less than total number of elements
i. Print the array elements

Step 3: Inside the main function
a. Create an integer array with necessary size
b. Get the total number of elements
c. Read the array elements one by one
d. Call the sort() function by passing array and no. of elements as arguments

## PROGRAM:

```
#include<stdio.h>
void sorting(int a[],int n)
{
int i,j,t;
for(i=0;i<n-1;i++)
{
for(j=i+1;j<n;j++)
{
if(a[i]>a[j])
{
t=a[i]; a[i]=a[j]; a[j]=t;
}
}
}
printf("Array Elemets before sorting\n");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
}
void main()
{
int i,a[10],n;
printf("Enter total no. of elements\n");
scanf("%d",&n);
printf("Enter Array Elements one by one\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
printf("Array Elemets before sorting\n");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
printf("\n");
sorting(a,n);
}
```


## OUTPUT:

Enter total no. of elements
6
Enter Array Elements one by one
21
2
9

Array Elemets before sorting
2129453011
Array Elemets after sorting
2911213045

## Result

Thus the program to sort the given array of elements using functions is executed successfully and its inputs and outputs are verified.

Ex.No: 7.d
FUNCTIONS WITH ARGUMENTS \& RETURN TYPE
Date:

## AIM:

To write a c program to find the smallest element of given array of elements using functions.

## ALGORITHM:

Step 1: Create a function small()
Step 2: Inside the function
a. Store the 0th index value into base $b$. Initialize a index to 1
c. Compare the array index value with base
d. If the array index value is smaller than base store the array index value into base
e. Increment the index by 1
f. Repeat the steps c \& e until index reaches total no. of elements
g. Return the base value

Step 3: Inside the main function
a. Create an integer array with necessary size b. Get the total number of elements
c. Read the array elements one by one
d. Call the small() function by passing array and no. of elements as arguments

## PROGRAM:

\#include<stdio.h>
int small(int a[],int n)
\{
int s,i; s=a[0]; for $(i=0 ; i<n ; i++)$
\{ if(a[i]<s) s=a[i];
\}
return s;

```
}
int main()
{
int i,a[10],n,s;
printf("Enter total no. of elements\n");
scanf("%d",&n);
printf("Enter Array Elements one by one\n");
for(i=0;i<n;i++) scanf("%d",&a[i]); printf("Array
Elemets:\n"); for(i=0;i<n;i++)
printf("%d\t",a[i]); printf("\n");
s=small(a,n);
printf("The Smallest element of given array is %d",s);
}
```


## OUTPUT:

Enter total no. of elements
5
Enter Array Elements one by one
1
98
2
66
0
Array Elemets:
1982660
The Smallest element of given array is 0

## Result

Thus the program to find the smallest element of given array of elements using functions is executed successfully and its inputs and outputs are verified.

Ex.No: 7.e
PROGRAM USING RECURSIVE FUNCTION
Date:

## AIM:

To write a c program to calculate the factorial of a given number

## ALGORITHM:

Step 1: Get the number
Step 2: Call the function fact by passing number as an argument
Step 3: Inside the fact()
a. If the received value is 0 or 1 than return 1
b. If the received value is not equal to 0 or 1 Multiply the value with return value of fact by
passing value -1 as an argument
c. Return the above result

Step 4: Print the result by receiving the return value of fact()

## PROGRAM:

\#include<stdio.h>
int factorial(int n)
\{
if( $n==0| | n==1$ )
return 1;
else
return ( $n$ *factorial( $n-1$ ));
\}
main()
\{
int n ;
printf("\nEnter a Number\n");
scanf("\%d",\&n);
printf("\nThe factorial of \%d is \%d\n",n,factorial(n));
\}

## OUTPUT:

Enter a Number 6
The factorial of 6 is 720

## Result

Thus the program to calculate the factorial of a given number using recursive function is executed successfully and its inputs and outputs are verified.

Ex.No: 7.e
PROGRAM TO DISPLAY ADDRESS OF A VARIABLE USING POINTERS
Date:

## AIM:

To write a C program to display address of a variable using pointers

## ALGORITHM:

1. Start
2. Declare variables and pointer variables
3. Assign the address of a variable to pointer variable
4. Display the value of variable
5. Display the address of the variable
6. Stop the program

## PROGRAM

\#include<stdio.h>
\#include<conio.h>
void main()
\{
int $x=5$;
int *a;
$a=\& x ;$
printf("\n The Value of $\left.x=\% d^{\prime}, * a\right)$;
printf("\n The Address of $x=\% u ", a)$;
\}

## OUTPUT:

The Value of $x=5$
The Address of $x=182261628$

## Result

Thus the program to display address of variable using pointers is executed successfully and its inputs and outputs are verified.

## Ex.No: 7.e

## PROGRAM TO FIND AREA OF CIRCLE USING POINTERS

## Date:

## AIM:

To write a C Program to find area of circle using pointers

## ALGORITHM:

1. Start
2. Declare required variables and pointer variables
3. Get the radius of the circle
4. Assign the address of variable having radius value to a pointer
5. Calculate radius of the circle using that pointer variable
6. Display the area
7. Stop

PROGRAM:
\#include<stdio.h>
\#include<conio.h>
void main()
\{
float r, area, pi=3.14;

```
    float *radius;
    radius=&r;
    printf("Enter the radius:");
    scanf("%f", &r);
    area=pi*(*radius)*(*radius);
    printf("The area of circle is:%f", area);
    getch();
}
```


## OUTPUT:

```
Enter the radius : 2.5
The area of circle is : 19.625000
```


## Result

Thus the program to find area of circle using pointers is executed successfully and its inputs and outputs are verified.

## Ex.No: 7.f

DYNAMIC MEMORY ALLOCATION USING POINTERS

## Date:

## AIM:

To write a C program to implement dynamic memory allocation using pointers

## ALGORITHM:

1. Start
2. Declare required variables
3. Declare required pointer variables
4. Allocate memory dynamically for the declared pointer variable using malloc() function
5. Display the value and memory address pointed by the pointer.

## PROGRAM:

\#include<stdio.h>
\#include<malloc.h>
\#include<conio.h>
void main()
\{

```
    int *fp;
    int val;
    fp=(int *)malloc(10);
    clrscr();
    printf("Enter a integer value : ");
    scanf("%d",fp);
```

printf("The address of pointer in memory is : \%u",fp);
printf(" $\backslash \mathrm{nThe}$ value stored in memory is : \%d", *fp);
getch();
\}

## OUTPUT:

Enter a integer value : 4
The address of pointer in memory is : 39772176
The value stored in memory is : 4

## Result

Thus the program to implement dynamic memory allocation is executed successfully and its inputs and outputs are verified.

Ex.No: 8.a
IMPLEMENTATION OF STUDENT RECORD USING STRUCTURES

## Date:

## AIM:

To write a c program to implement student record using structures.

## ALGORITHM:

Step 1: Create a structure student with roll no, name, dept and 3 marks as fields
Step 2: Create a structure variable
Step 3: Read the input for student details
Step 4: Calculate the average of student by using 3 marks
Step 5: Print the structure elements using structure variable

## PROGRAM:

\#include<stdio.h>
struct student
\{
int rno,m1,m2,m3;
float avg;
char name[20],dept[10];
\};
void main()
\{
struct student s;
clrscr();
printf("Enter the Student Details:\n");
printf("Enter the Student roll no:\n");
scanf("\%d",\&s.rno);

```
    printf("Enter the Student Name:\n");
    scanf("%s",&s.name);
    printf("Enter the Student Dept:\n");
    scanf("%s",&s.dept);
    printf("Enter the 3 marks:\n");
    scanf("%d%d%d",&s.m1,&s.m2,&s.m3);
    s.avg=(s.m1+s.m2+s.m3)/3;
printf("The Student Average is :%f\n",s.avg);
getch();
}
```


## OUTPUT:

Enter the Student Details:
Enter the Student roll no:12
Enter the Student Name: Kumar
Enter the Student Dept: CSE
Enter the Student marks:
40
18
90
The Student Average is :49.000000

## Result

Thus the program to implement student record using structures is executed successfully and its inputs and outputs are verified.

Ex.No: 8.b
IMPLEMENTATION OF ARRAY OF STRUCTURES

## Date:

## AIM:

To write a C program to implement array of structures.

## ALGORITHM:

Step 1: Create a structure student with roll no, name, dept and 3 marks as fields
Step 2: Create a structure variable with necessary size
Step 3: Read the total number of students
Step 3: Read the structure details for all students
Step 4: Calculate the average of students by using 3 marks
Step 5: Write a function to print the student details if roll no is given
Step 6: Inside the function
a. Initialize the index to 0
b. Compare the roll no. with structure index roll no.
c. If they are same print the student details
d. Increment the index by 1
e. Repeat the steps $b$ to $d$ until index reaches the total no. of students

## PROGRAM:

## \#include<stdio.h>

```
struct student
```

\{
int rno,m1,m2,m3;
float avg;
char name[20],dept[10];
\};
void find_student(int a,struct student s[],int n)
\{
int i ;
printf("The Student Detail for roll no: \%d $\backslash n$ ", a);
for(i=0;i<n;i++)
\{
if(s[i].rno==a)
\{
printf("\%s $\backslash t \% s \backslash t \% d \backslash t \% d \backslash t \% d \backslash t \% f \backslash n ", s[i] . n a m e, s[i] \cdot d e p t, s[i] \cdot m 1, s[i] . m 2, s[i] . m 3, s[i] \cdot a v g) ;$
break;
\}
\}
\}
void main()
\{
int i,n,rno;
struct student s[10];
clrscr();
printf("Enter total no. of Students\n");
scanf("\%d",\&n);
for (i=0;i<n;i++)
printf("Enter the Student \%d Details: ${ }^{n}$ ",(i+1));
printf("Enter the roll no:\n");
scanf("\%d",\&s[i].rno);
printf("Enter the Name:\n");
scanf("\%s",\&s[i].name);
printf("Enter the Dept:\n");
scanf("\%s",\&s[i].dept);
printf("Enter the 3 marks:\n");
scanf("\%d\%d\%d",\&s[i].m1,\&s[i].m2,\&s[i].m3);
$\mathrm{s}[\mathrm{i}] \cdot \mathrm{avg}=(\mathrm{s}[\mathrm{i}] \cdot \mathrm{m} 1+\mathrm{s}[\mathrm{i}] \cdot \mathrm{m} 2+\mathrm{s}[\mathrm{i}] \cdot \mathrm{m} 3) / 3$;

```
    }
    printf("Enter the roll no to find details:\n");
    scanf("%d",&rno);
    find_student(rno,s,n);
    getch();
}
```


## OUTPUT:

Enter total no. of Students: 2
Enter the Student 1 Details:
Enter the roll no:12
Enter the Name: Noor
Enter the Dept:cse
Enter the 3 marks:
45
67
88
Enter the Student 2 Details:
Enter the roll no:13
Enter the Name: Nilo
Enter the Dept:cse
Enter the 3 marks:
77
89
67
Enter the roll no to find details:13
The Student Detail for roll no : 13
Nilo cse 77896777.000000

## Result

Thus the program to implement array of structures is executed successfully and its inputs and outputs are verified.

Ex.No: 9.a
FILE MANIPULATION USING getc() AND putc() FUNCTIONS

## Date:

## AIM:

To write a C program to manipulate file using getc() and putc() functions

## ALGORITHM:

1. Start
2. Declare the file pointer *f1
3. Open the file in write mode.
4. Write the content of file using putc() function
5. Close the file
6. Open the file in read mode
7. Read the content in file using getc() function
8. Close the file.

## PROGRAM:

```
#include <stdio.h>
```

void main()
\{
FILE *f1;
char c;
f1= fopen("INPUT", "w"); /* open file for writing */
while((c=getchar()) != EOF) /*get char from keyboard until CTL-Z*/
putc(c,f1); /*write a character to INPUT */
fclose(f1); /* close INPUT */
f1=fopen("INPUT", "r"); /* reopen file */
while((c=getc(f1))!=EOF) /*read character from file INPUT*/
printf("\%c", c);/* print character to screen */
fclose(f1);
\}
OUTPUT:
Noornilo Nafees $\mathrm{ctrl}^{\wedge} \mathrm{z}$
Noornilo Nafees

## Result

Thus the program to manipulate file using getc() and putc() functions is executed successfully and its inputs and outputs are verified.

## Ex.No: 9.b

FILE MANIPULATION USING fprintf() AND fscanf() FUNCTIONS

## Date:

## AIM:

To write a C program to manipulate file using fprintf() and fscanf() functions

## ALGORITHM:

1. Start
2. Declare the file pointer
3. Open the file in write mode.
4. Write the content of file using fprintf() function
5. Close the file
6. Open the file in read mode
7. Read the content in file using fscanf() function
8. Close the file.

## PROGRAM:

## //Program for file manipulation using fprintf()

\#include<stdio.h>
void main()
\{
FILE *fp;
int roll;
char name[25];
float marks;
char ch;
fp = fopen("file.txt","w"); //Statement 1
if(fp == NULL)
\{
printf("\nCan't open file or file doesn't exist.");
exit(0);
\}
do
\{
printf("\nEnter Roll : ");
scanf("\%d",\&roll);
printf("\nEnter Name : ");
scanf("\%s",name);
printf(" $\operatorname{n}$ nEnter Marks: ");
scanf("\%f",\&marks); fprintf(fp,"\%d\%s\%f",roll,name,marks);
printf(" $\backslash \mathrm{nDo}$ you want to add another data ( $\mathrm{y} / \mathrm{n}$ ) : ");
ch = getche();
\}while(ch=='y' || ch=='Y');
printf("\nData written successfully...");
fclose(fp);
\}

## OUTPUT:

Enter Roll : 1
Enter Name : Nilo
Enter Marks : 78.53
Do you want to add another data ( $\mathrm{y} / \mathrm{n}$ ) : y
Enter Roll : 2
Enter Name : Nafees
Enter Marks : 89.62
Do you want to add another data ( $\mathrm{y} / \mathrm{n}$ ) : n
Data written successfully...

## //Program for file manipulation using fscanf()

\#include<stdio.h>
void main()
\{
FILE *fp;
char ch;
fp = fopen("file.txt","r"); //Statement 1
if(fp == NULL)
\{ printf("\nCan't open file or file doesn't exist."); exit(0);
\}
printf("\nData in file...\n");
while((fscanf(fp,"\%d\%s\%f",\&roll,name,\&marks))!=EOF) //Statement 2
\{
printf("\n\%d\t\%s\t\%f",roll,name,marks);
\}
fclose(fp);
\}
OUTPUT :
Data in file...
1 Nilo 78.53
2 Nafees 89.62

## Result

Thus the program to manipulate file using fprintf() and fscanf() functions is executed successfully and its inputs and outputs are verified.

