# **SRI VENKATESHWARA COLLEGE OF ENGINEERING**

Pennalur, Sripreumbudur TK- 602 117



**RECORD NOTE BOOK**

*Certified to be a Bonafide Record Work of*

Name : <your name>

University Register Number: 212723 <register\_no>

Laboratory : Programming and Problem Solving Lab

Semester : 1

Academic Year : 2023 - 24

Branch : <branch>

Date :

Laboratory Incharge

Submitted for the Practical Examinations held on ( / / ).

Internal Examiner External Examiner

Programming and Problem Solving Laboratory

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | **S.NO** | **DATE** | **NAME OF THE EXPERIMENT** | **DATE OF SUBMISSION** | **MARKS** |  |
|  | 1 |  | USAGE OF BASIC LINUX COMMANDS |  |  |  |
|  | 2 |  | C PROGRAMMING USING SIMPLE STATEMENTS AND EXPRESSIONS |  |  |  |
|  | 3 |  | SCIENTIFIC PROBLEMS SOLVING USING DECISION MAKING |  |  |  |
|  | 4 |  | SCIENTIFIC PROBLEM SOLVING USING LOOPING |  |  |  |
|  | 5 |  | PROGRAMMING FOR 1D AND 2D ARRAYS |  |  |  |
|  | 6 |  | SOLVING PROBLEMS USING STRINGS |  |  |  |
|  | 7 |  | PROGRAMS TO ILLUSTRATE STRUCTURES AND UNION |  |  |  |
|  | 8 |  | PROGRAMS TO ILLUSTRATE POINTERS |  |  |  |
|  | 9 |  | PROGRAMS TO ILLUSTRATE USER- DEFINED FUNCTIONS |  |  |  |
|  | 10 |  | FILE HANDLING IN C |  |  |  |
|  | 11 |  | ENUMERATED DATATYPES, MACROS AND STRORAGE CLASSES |  |  |  |
|  |  |  |  |  |  |  |

Experiment No.:1 Date:

# **USAGE OF BASIC LINUX COMMANDS**

**AIM:**

**COMMANDS AND THEIR DESCRIPTION:**

**RESULT:**

Thus all the Linux commands and their usages has been applied.

Experiment No.: 2 Date:

# **C PROGRAMMING USING SIMPLE STATEMENTS AND EXPRESSIONS**

## **0. *“Print your <name>, I am an Engineer”*. Using C program**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

main()

{

    printf("My name is Antony  Xavio Immanuel, I am an Engineer.");

}

**OUTPUT:**

My name is Antony  Xavio Immanuel, I am an Engineer.

## **1. Print your details Name, age, Gender (M/F), Address, Cut-Off (in HSC)**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

main()

{

    printf("Name: Antony Xavio Immanuel \n Age:18 \n Gender: Male \n Address: Kottivakkam, Chennai-41 \n Cutoff[HSC]:181\n");

}

**OUTPUT:**

Name: Antony Xavio Immanuel

Age:18

Gender: Male

Address: Kottivakkam, Chennai-41

Cutoff[HSC]:181

## **2. Read your details through keyboard and display them**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

    int age;

    char name[20];

    char gender;

    float cutoff;

    char address[50];

    printf("\nEnter your gender,age,name,cutoff and address\n");

    scanf("%c%d%s%f", &gender, &age, &name, &cutoff);

    scanf("%s", &address);

    printf("\nName: %s \n Age: %d \n Gender: %c \n Address: %s \n

Cutoff[HSC]: %f\n", name, age, gender, address, cutoff);

}

**OUTPUT:**

Enter your gender,age,name,cutoff and address

M

18

Immanuel

181

Kottivakkam,Chennai-41

Name: Immanuel

Age: 18

Gender: M

Address: Kottivakkam,Chennai-41

Cutoff[HSC]: 181.000000

## **3. Kumar has Rs.x and Sheela has Rs.Y, exchange their amounts using interactive C program.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

    int kumar, sheela, temp = 0;

    printf("Enter the amt which Kumar and Sheela have:\n");

    scanf("%d%d", &kumar, &sheela);

    temp = kumar;

    kumar = sheela;

    sheela = temp;

    printf("The amt which Kumar and Sheela have after exchanging is:\n");

    printf("Kumar:%d", kumar);

    printf("\nSheela:%d", sheela);

}

**OUTPUT:**

Enter the amt which Kumar and Sheela have:

700

850

The amt which Kumar and Sheela have after exchanging is:

Kumar:850

Sheela:700

## **4. Calculate the daily wage of a labour taking into account the pay per hour, number of hours worked, TA, and DA.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

    int pph,hw;

    printf("\tDAILY WAGE CALCULATOR\nTA set at 30%% of pay per day\nDA set at 15%% of pay per day\n\nEnter the pay per hour:");

    scanf("%d",&pph);

    printf("\nEnter the number of hours worked:");

    scanf("%d",&hw);

    int total = pph\*hw;

    int ta= 0.3\*total;

    int da= 0.15\*total

    printf("\n\nBase amount=%d\nTransport allowance=%d\nDearness allowance=%d\nTotal=%d",total,ta,da,total+da+ta);

}

**OUTPUT:**

DAILY WAGE CALCULATOR

TA set at 30% of pay per day

DA set at 15% of pay per day

Enter the pay per day:500

Enter the number of hours worked:8

Base amount=4000

Transport allowance=1200

Dearness allowance=600

Total=5800

## **5. Box 1 has M apples and Box 2 has N apples, exchange the apples in Box 1 and Box 2. (Swap without temp).**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

    int b1, b2;//eg: b1=5,b2=3

    printf("Enter the number of apples in box1 and box2 have:\n");

    scanf("%d%d", &b1, &b2);

    b1 = b1 + b2; //b1=5+3=8

    b2 = b1 - b2;//b2=8-3=5

    b1 = b1 - b2;//b1=8-5=3

    printf("The number of apples box1 and box2 after exchanging is\n");

    printf("b1:%d", b1);

    printf("\nb2:%d", b2);

}

**OUTPUT:**

Enter the number of apples in box1 and box2 have:

10

6

The number of apples box1 and box2 after exchanging is

b1:6

b2:10

## **6. Display the sum of ASCII values of vowels in English letters.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

    int sum = 0;

    char vowels[] = {'a', 'e', 'i', 'o', 'u'};

    for (int i = 0; i < 5; i++)

    {

        int m = (int)(vowels[i]);

        sum += m;

        printf("%c\t%d\n", vowels[i], vowels[i]);

    }

    printf("\n\nsum of ASCII values of vowels (lowercase) is: %d", sum);

}

**OUTPUT:**

a 97

e 101

i 105

o 111

u 117

sum of ASCII values of vowels (lowercase) is: 531

## **7. Read a complex number. Display the number and calculate its modulus value.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <math.h>

void main()

{

int a[2];

printf("\n\nMODULUS OF A COMPLEX NUMBER”);

printf(“\nEnter the real value of complex no.:");

scanf("%d", &a[0]);

printf("\nEnter the imaginary value:");

scanf("%d", &a[1]);

printf("\nThe given complex number is: (%d)+(%d)i",a[0],a[1]);

int mod= pow(a[0],2) + pow(a[1],2);

printf("\nThe modulus is:%f",sqrt(mod));

}

**OUTPUT:**

MODULUS OF A COMPLEX NUMBER

Enter the real value of complex no.:2

Enter the imaginary value:9

The given complex number is: (2)+(9)i

The modulus is:9.219544

**RESULT:**

Thus all C programs with simple statement and expressions has been executed successfully.

Experiment No.: 3 Date:

# **SCIENTIFIC PROBLEMS SOLVING USING DECISION MAKING**

## **1. Ram, Balaji, and Kumar are siblings. Find the youngest and oldest.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

int x, y, z;

printf("Enter the ages of Ram, Balaji and Kumar: \n");

scanf("%d%d%d", &x, &y, &z);

char max[20] = "", min[20] = "";

(x > y) ? (x > z) ? strcpy(max, "Ram") : strcpy(max, "Kumar") :

(y > z) ? strcpy(max, "Balaji"):strcpy(max, "Kumar");

(x < y) ? (x < z) ? strcpy(min, "Ram") : strcpy(min, "Kumar") :

(y < z) ? strcpy(min, "Balaji"): strcpy(min, "Kumar");

printf("\n %s is the oldest", max);

printf("\n %s is the youngest", min);

}

**OUTPUT:**

Enter the ages of Ram, Balaji and Kumar:

10

22

7

Balaji is the oldest

Kumar is the youngest

## **2. Calculate the ESE marks of a subject.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

printf("\t\tESE CALCULATOR");

printf("\n\n\t\tINTERNAL MARKS");

int cat1, cat2, cat3, agn1, agn2, agn3;

printf("\nEnter marks of CAT1, CAT2, CAT3\n");

scanf("%d%d%d", &cat1, &cat2, &cat3);

printf("\nEnter marks of Assignment1, Assignment2, Assignment3\n");

scanf("%d%d%d", &agn1, &agn2, &agn3);

float internal =(cat1+cat2+cat3)\*0.7+(agn1+agn2+agn3)\*0.3;

int external, subject\_type = 0;

printf("\n\nEXTERNAL MARK\nEnter external(100) mark:");

scanf("%d", &external);

printf("\n\nEnter 1 for Theory \nEnter 0 for Practical\n");

scanf("%d", &subject\_type);

float total =(subject\_type == 1)?0.4\*internal+0.6\*external:

0.4\* external + 0.6 \* internal;

printf("\n\nESE marks:%d. \n\nProgram terminating....\n\n", total);

}

**OUTPUT:**

ESE CALCULATOR

INTERNAL MARKS

Enter marks of CAT1, CAT2, CAT3

50

43

38

Enter marks of Assignment1, Assignment2, Assignment3

47

50

50

EXTERNAL MARK

Enter external(100) mark:97

Enter 1 for Theory

Enter 0 for Practical

1

ESE marks:112.520004.

Program terminating....

## **3. Print the number of days in a month using if-else statement**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

printf("\n\n\t Number of days in a month of a year calculator\n");

int month, days = 0;

printf("\n Enter month:");

scanf("%d", &month);

if(month==2){

int year;

printf("\n Enter the year:");

scanf("%d", &year);

if((year%400==0 && year%100==0)||(year%4==0 && year%100!=0))

{days=29;}

else{days=28;}

} else if(month==4||month==6||month==9||month==11)

{days=30;}

else

{days=31;}

}

**OUTPUT:**

Number of days in a month of a year calculator

Enter month:2

Enter the year:2004

Number of days in the month 2 is 29

## **4. *‘Solar System’* is a program that reads a number and prints the respective planet using simple switch.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main(){

int x;

printf("\n\n\tSOLAR SYSTEM\nEnter the numerical order of the planet:");

scanf("%d", &x);

printf("\n");

switch (x){

case 1:

printf("Mercury");

break;

case 2:

printf("Venus");

break;

case 3:

printf("Earth");

break;

case 4:

printf("Mars");

break;

case 5:

printf("Jupiter");

break;

case 6:

printf("Saturn");

break;

case 7:

printf("Uranus");

break;

case 8:

printf("Neptune");

break;

default: printf("That planet does not exist");

} }

**OUTPUT:**

SOLAR SYSTEM

Enter the numerical order of the planet:8

Neptune

SOLAR SYSTEM

Enter the numerical order of the planet:1

Mercury

SOLAR SYSTEM

Enter the numerical order of the planet:3

Earth

SOLAR SYSTEM

Enter the numerical order of the planet:6

Saturn

## **5. Develop a ‘Kids Laptop’ using switch case clustering in C program.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

char x;

printf("\n\n\tKIDS LAPTOP\nEnter the alphabet: ");

scanf("%c", &x);

printf("\n\n");

switch (x){

case 'A':

case 'a':

printf("Apple");

break;

case 'B':

case 'b':

printf("Ball");

break;

case 'C':

case 'c':

printf("Cat");

break;

case 'D':

case 'd':

printf("Doll");

break;

case 'E':

case 'e':

printf("Elephant");

break;

case 'F':

case 'f':

printf("Fox");

break;

case 'G':

case 'g':

printf("Grapes");

break;

case 'H':

case 'h':

printf("Horse");

break;

default:

printf("Try again..");

}

}

**OUTPUT:**

KIDS LAPTOP

Enter the alphabet: c

Cat

**RESULT:**

Thus all C programs related to scientific problem solving using decision

making have been executed successfully.

Experiment No.: 4 Date:

# **SCIENTIFIC PROBLEM SOLVING USING LOOPING**

## **1. Write a program that prints the numbers from 1 to 100 and checks for *“fizz buzz”***

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

printf("\n\tFIZZBUZZ\n");

for (int i = 1; i <= 100; i++)

{

if (i % 3 == 0 && i % 5 == 0){printf("%d-FizzBuzz \n", i);}

else if (i % 3 == 0){printf("%d-Fizz \n", i);}

else if (i % 5 == 0){printf("%d-Buzz \n", i);}

else{printf("%d \n", i);}

}

}

**OUTPUT:**

FIZZBUZZ

1

2

3-Fizz

4

5-Buzz

6-Fizz

……………

15-FizzBuzz

……………

100

## **2. An astrologer program to calculate the day’s fortune.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main(){

int num, val, sum = 0;

printf("\n\tFORTUNE CALCULATOR");

printf("\nEnter date of birth (eg:24092005 for 24/09/2005):");

scanf("%d", &num);

for (int i = 0; i < 2; i++){

sum = 0;

while (num != 0){

val = num % 10;

sum += val;

num /= 10;}

num = sum;}

printf("Fortune number is:%d", sum);

if (sum <= 5) {printf("\n\nDay is Excellent");}

else if (sum <= 8) {printf("\n\nDay is Fair");}

else {printf("\n\nDay is Good");}

}

**OUTPUT:**

FORTUNE CALCULATOR

Enter date of birth (eg:24092005 for 24/09/2005):17111971

Fortune number is: 10

Day is Good

## **3.Program to convert numbers into their respective text ( 81- eight one)**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main(){

int num, val, c, r = 0;

char a2[10][6] = {"Zero", "One", "Two", "Three", "Four", "Five", " Six", "Seven", "Eight", "Nine"};

printf("\n\n\tNum to Text converter\nEEnter the ESE mark: ");

scanf("%d", &num);

while (num != 0){

c = num % 10;

r = r \* 10 + c;

num /= 10;

}

while (r != 0){

val = r % 10;

printf("%s ", a2[val]);

r /= 10;

}

}

**OUTPUT:**

Num to Text converter

Enter the ESE mark:96

Nine Six

## **4. Program to print the following patterns**

|  |  |  |
| --- | --- | --- |
| 1)  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5 | 2)  1  2 3  4 5 6  7 8 9 10  11 12 13 14 15 | 3)  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \* |

**1)**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

for (int i = 1; i <= 5; i++)

{

for (int j = 1; j <= i; j++)

{

printf("%d\t", i);

}

printf("\n");

}

}

**OUTPUT:**

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

**2)**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int n = 1;

for (int i = 1; i <= 5; i++)

{

for (int j = 1; j <= i; j++)

{

printf("%d\t", n);

n++;

}

printf("\n");

}

}

**OUTPUT:**

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

**3)**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main() {

int n=5;n++;

printf("\n\n");

for (int k = 1; k <= n; k++)

{

for (int c = 0; c < n - k; c++) {printf(" ");}

for (int c = 1; c <=k; c++) {printf(" \*");}

}

}

**OUTPUT:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**RESULT:**

Thus all C programs related to scientific problem solving with looping has been executed successfully.

Experiment No.: 5 Date:

# **PROGRAMMING FOR 1D AND 2D ARRAYS**

## **1. Prof. Siva awards marks for N questions in the range 0-16. Support him in finding the total.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int n, tot = 0;

printf("Enter the number of questions: ");

scanf("%d", &n);

int a[n];

for (int i = 0; i < n; i++)

{

printf("Enter the mark:");

scanf(" %d", &a[i]);

if (a[i] > 16)

{

printf("Re-");

i--;

continue;

}

tot += a[i];

if (tot >= 100)

{

printf("\nTotal exceeded or reached 100 .. terminating..");

break;

}

}

printf("\nThe total is:%d", tot);

}

**OUTPUT:**

Enter the number of questions: 10

Enter the mark:16

Enter the mark:9

Enter the mark:10

Enter the mark:21

Re-Enter the mark:2

Enter the mark:11

Enter the mark:18

Re-Enter the mark:8

Enter the mark:7

Enter the mark:14

Enter the mark:12

Enter the mark:5

The total is:94

## **2. Dr. Sudha marks daily attendance of N students. Help her to find the number of present and absentees for a day.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int n, p = 0, a = 0;

printf("\nEnter the number of students:");

scanf("%d", &n);

char c[n];

for (int i = 1; i <= n; i++)

{

printf("student %d - ", i);

scanf(" %c", &c[i]);

(c[i] == 'P' || c[i] == 'p') ?++p:++a;

}

printf("\nThe number of students present is: %d", p);

printf("\nThe number of students absent is: %d", a);

}

**OUTPUT:**

Enter the number of students:5

student 1 - p

student 2 - a

student 3 - a

student 4 - p

student 5 - p

The number of students present is: 3

The number of students absent is: 2

## **3. Prof. Vimal is interested in forming project teams as a triplet of every alternate student in his class. Write a program for the same .**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int n;

printf("\n\nEnter the number of students in the class:");

scanf("%d", &n);

if (n % 3 == 0){

n /= 3;

int a[n][3];

for (int i = 0; i < 3; i++){

for (int j = 0; j < n; j++){

printf("\nEnter roll:");

scanf(" %d", &a[j][i]);

}

}

printf("\n");

for (int i = 0; i < n; i++){

printf("group %d:\n", i + 1);

for (int j = 0; j < 3; j++){

printf("%d\t", a[i][j]);

}

printf("\n\n");

}

}

else{

printf("Terminating program, as class size is not multiple of 3");

}

}

**OUTPUT:**

Enter the number of students in the class:6

Enter roll:10

Enter roll:11

Enter roll:12

Enter roll:13

Enter roll:14

Enter roll:15

group 1:

10 12 14

group 2:

11 13 15

## **4. Write a program to calculate the Mean, Variance, and SD of a N values**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <math.h>

void main()

{

float m=0,v=0,sd=0;

int n;

printf("\nEnter the number of values:");

scanf("%d", &n);

int a[n];

for (int i = 0; i < n; i++)

{

printf("\nEnter %d val:");

scanf(" %d", &a[i]);

m+=a[i];

}

m/=n;

for(int i=0;i<n;i++)

{

v+= (pow((a[i]-m),2)/n);

}

sd=sqrt(v);

printf("\n\nThe mean of the give array of values is:%f",m);

printf("\n\nThe variance of the give array of values is:%f",v);

printf("\n\nThe SD of the give array of values is:%f",sd);

}

**OUTPUT:**

Enter the number of values:5

Enter 1 val:18

Enter 2 val:22

Enter 3 val:35

Enter 4 val:73

Enter 5 val:11

The mean of the give array of values is:31.799999

The variance of the give array of values is:485.359985

The SD of the give array of values is:22.030888

## **5.** **Read a square a and an element e in it, write a program to print the elements adjacent to e.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int a[3][3];

printf("\nTo find adjacent element in 3x3 a\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

printf("\nEnter value for a[%d][%d]:", i+1, j+1);

scanf("%d", &a[i][j]);

}

}

int e,chk=0;

printf("\nEnter the element to search adjacent for:");

scanf(" %d", &e);

for (int i = 0; i < 3; ++i)

{

for (int j = 0; j < 3; ++j)

{

if (a[i][j] == e)

{

chk=1;

if (i > 0)

printf("Above: %d\n", a[i - 1][j]);

if (i < 2)

printf("Below: %d\n", a[i + 1][j]);

if (j > 0)

printf("Left: %d\n", a[i][j - 1]);

if (j < 2)

printf("Right: %d\n", a[i][j + 1]);

if (i == j)

{

printf("Diagonal: %d,%d,%d,%d",

a[0][0], a[0][2], a[2][0], a[2][2]);

break;

}

if (i + j == 2 || i + j == 0)

printf("Diagonal: %d\n", a[1][1]);

if (i + j == 1)

printf("Diagnonal: %d, %d", a[j][i], a[i + 1][j + 1]);

if (i + j == 3)

printf("Diagnonal: %d, %d", a[j][i], a[i - 1][j - 1]);

break;

}

}

}

(chk==0)? printf("\nElement not found, terminating the program")

:printf("\n");

}

**OUTPUT:**

To find adjacent element in 3x3 a

Enter value for a[1][1]:1

Enter value for a[1][2]:2

Enter value for a[1][3]:3

Enter value for a[2][1]:4

Enter value for a[2][2]:5

Enter value for a[2][3]:6

Enter value for a[3][1]:7

Enter value for a[3][2]:8

Enter value for a[3][3]:9

Enter the element to search adjacent for:2

Below: 5

Left: 1

Right: 3

Diagnonal: 4, 6

## **6. Read the marks scored by M students in N subjects and, calculate the subject wise maximum marks, the total of each student, the row index of the topper student, find the grand total of all marks of all students.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int m, n;

printf("\nEnter the number of students:");

scanf("%d", &m);

printf("\nEnter the number of subjects:");

scanf("%d", &n);

int a[m][n], sub[n], stud[m], grand\_sum = 0;

for (int i = 0; i < m; i++)

{

printf("\nEnter marks for student %d:\n", i + 1);

for (int j = 0; j < n; j++)

{

printf("\tEnter the mark for subject %d:", j + 1);

scanf("%d", &a[i][j]);

grand\_sum += a[i][j];

sub[j] = a[0][j];

stud[i] = 0;

}

printf("\n");

}

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

{

if (a[i][j] > sub[j])

sub[j] = a[i][j];

stud[i] += a[i][j];

}

}

for(int j=0;j<n;j++)

printf("\nMaximum mark in Subject %d is %d",j+1,sub[j]);

printf("\n\n");

int topper = stud[0], index = 0;

for (int i = 0; i < m; i++)

{

if (stud[i] > topper)

{

topper = stud[i];

index = i;

}

printf("\nTotal marks of Student %d is : %d",i+1,stud[i]);

}

printf("\n\nTop score of the class is:%d\nRow index of the topper

student is:%d", topper, ++index);

printf("\n\nGrand total of all marks: %d",grand\_sum);

}

**OUTPUT:**

Enter the number of students:3

Enter the number of subjects:2

Enter marks for student 1:

Enter the mark for subject 1:90

Enter the mark for subject 2:85

Enter marks for student 2:

Enter the mark for subject 1:76

Enter the mark for subject 2:83

Enter marks for student 3:

Enter the mark for subject 1:99

Enter the mark for subject 2:92

Maximum mark in Subject 1 is 99

Maximum mark in Subject 2 is 92

Total marks of Student 1 is : 175

Total marks of Student 2 is : 159

Total marks of Student 3 is : 191

Top score of the class is:191

Row index of the topper student is:3

Grand total of all marks: 525

**RESULT:**

Thus the required output is printed.

## **7.** **Read the sales done by 2 salesmen in 5 months for 4 products and calculate the total sales.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main(){

int s1[4][5], s2[4][5], prod[4];

printf("\nTOTAL SALES CALCULATOR\n");

printf("Enter sales done by Salesman 1\n");

for (int i = 0; i < 4; i++){

for (int j = 0; j < 5; j++){

printf("\nEnter sale of product-%d on month-%d: ", i + 1, j + 1);

scanf("%d", &s1[i][j]);

}

}

printf("\n\nEnter sales done by Salesman 2\n");

for (int i = 0; i < 4; i++){

for (int j = 0; j < 5; j++){

printf("\nEnter sale of product-%d on month-%d: ", i + 1, j + 1)

scanf("%d", &s2[i][j]);

}

}

printf("Printing the total sales done for a product in a month:");

for (int i = 0; i < 4; i++){

for (int j = 0; j < 5; j++){

int prod = s1[i][j]+s2[i][j];

printf("\n\nSales of product %d, in the month %d is:%d"

,i+1,j+1, prod);

}

}

}

**OUTPUT:**

TOTAL SALES CALCULATOR

Enter sales done by Salesman 1

Enter sale of product-1 on month-1: 4

Enter sale of product-1 on month-2: 7

Enter sale of product-1 on month-3: 2

Enter sale of product-1 on month-4: 6

Enter sale of product-1 on month-5: 3

Enter sale of product-2 on month-1: 6

Enter sale of product-2 on month-2: 9

Enter sale of product-2 on month-3: 9

Enter sale of product-2 on month-4: 7

Enter sale of product-2 on month-5: 2

Enter sale of product-3 on month-1: 6

Enter sale of product-3 on month-2: 4

Enter sale of product-3 on month-3: 1

Enter sale of product-3 on month-4: 7

Enter sale of product-3 on month-5: 5

Enter sale of product-4 on month-1: 3

Enter sale of product-4 on month-2: 5

Enter sale of product-4 on month-3: 9

Enter sale of product-4 on month-4: 2

Enter sale of product-4 on month-5: 9

Enter sales done by Salesman 2

Enter sale of product-1 on month-1: 7

Enter sale of product-1 on month-2: 4

Enter sale of product-1 on month-3: 9

Enter sale of product-1 on month-4: 9

Enter sale of product-1 on month-5: 1

Enter sale of product-2 on month-1: 5

Enter sale of product-2 on month-2: 3

Enter sale of product-2 on month-3:4

Enter sale of product-2 on month-4: 5

Enter sale of product-2 on month-5: 8

Enter sale of product-3 on month-1: 7

Enter sale of product-3 on month-2: 2

Enter sale of product-3 on month-3: 8

Enter sale of product-3 on month-4: 5

Enter sale of product-3 on month-5: 1

Enter sale of product-4 on month-1: 9

Enter sale of product-4 on month-2: 3

Enter sale of product-4 on month-3: 3

Enter sale of product-4 on month-4: 3

Enter sale of product-4 on month-5: 6

Printing the total sales done for a product in a month:

Sales of product 1, in the month 1 is: 11

Sales of product 1, in the month 2 is: 11

Sales of product 1, in the month 3 is: 11

Sales of product 1, in the month 4 is: 15

Sales of product 1, in the month 5 is: 4

Sales of product 2, in the month 1 is: 11

Sales of product 2, in the month 2 is: 12

Sales of product 2, in the month 3 is: 13

Sales of product 2, in the month 4 is: 12

Sales of product 2, in the month 5 is: 10

Sales of product 3, in the month 1 is: 13

Sales of product 3, in the month 2 is: 6

Sales of product 3, in the month 3 is: 9

Sales of product 3, in the month 4 is: 12

Sales of product 3, in the month 5 is: 6

Sales of product 4, in the month 1 is: 12

Sales of product 4, in the month 2 is: 8

Sales of product 4, in the month 3 is: 12

Sales of product 4, in the month 4 is: 5

Sales of product 4, in the month 5 is: 15

## **8.** **A shopkeeper delivers rice bags to three shops, calculate the income from the three shops separately (Vector-Matrix Multiplication)**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int x[3][3] = {{100, 100, 100}, {200, 200, 200}, {300, 300, 300}};

int y[1][3] = {600, 600, 600};

int c[1][3] = {0, 0, 0};

for (int i = 0; i < 1; i++)

{

for (int j = 0; j < 3; j++)

{

for (int k = 0; k < 3; k++)

{

c[i][j] += (y[i][k] \* x[j][k]);

}

printf("\n Total income from Shop-%d is: %d", j + 1, c[i][j]);

}

}

}

**OUTPUT:**

Total income from Shop-1 is: 180000

Total income from Shop-2 is: 360000

Total income from Shop-3 is: 540000

**RESULT:**

Thus all C programs relate to 1D and 2D arrays have been executed successfully.

Experiment No.: 6 Date:

# **SOLVING PROBLEMS USING STRINGS**

## **0.** **Print your name by initializing and reading from keyboard using char \*s.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

void main()

{

char s[30];

printf("Enter your name: ");

scanf("%s",&s);

printf("\nYour name is: %s",s);

}

**OUTPUT:**

Enter your name: Antony

Your name is: Antony

## **2. Write a program to read a name and count the number of vowels in it.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

char s[30];

int c=0;

printf("\n\nEnter your name:");

scanf("%s", &s);

printf("\nYour name is: %s", s);

for (int i = 0; i <= strlen(s); i++)

{

if(s[i]=='A'||s[i]=='a'||s[i]=='E'||s[i]=='e'||s[i]=='I'||s[i]=='i'

||s[i]=='O'||s[i]=='o'||s[i]=='U'||s[i]=='u')

{

++c;

}

}

printf("\nThe number of vowels in your name is:%d",c);

}

**OUTPUT:**

Enter your name:Immanuel

Your name is: Immanuel

The number of vowels in your name is:4

## **3. Read the first name, middle name, and last name of a person and form a full name as the concatenation of them.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

char fn[15],mn[15],ln[15];

printf("\n\nEnter your first name:");

gets(fn);

printf("\nEnter your middle name:");

gets(mn);

printf("\nEnter your last name:");

gets(ln);

char name[100]=" ";

strcat(name,fn);

strcat(name," ");

strcat(name,mn);

strcat(name," ");

strcat(name,ln);

printf("\n\nFull name is:");

puts(name);

}

**OUTPUT:**

Enter your first name:Antony

Enter your middle name:Xavio

Enter your last name:Immanuel

Full name is: Antony Xavio Immanuel

## **4.** **Write a C program that calculates the strength of the string.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main(){

char s[30];

printf("\n\nEnter your name:");

fgets(s,sizeof(s),stdin);

printf("\nYour name is: %s", s);

printf("\nEncrypted name is:\n");

for (int i = 0; i <= strlen(s); i++){

if (s[i] == 'A' || s[i] == 'a' || s[i] == 'E' || s[i] == 'e' ||

s[i] == 'I' || s[i] == 'i' || s[i] == 'O' || s[i] == 'o' ||

s[i] == 'U' || s[i] == 'u'){

printf("%c", s[i]);

}

else{

char x= ++s[i];

printf("%c",x);

}

}

}

**OUTPUT:**

Enter your name:Christopher

Your name is: Christopher

Encrypted name is:

Disituoqies

## **5.** **Write a C program to design a encryption technology that replaces every non-vowel.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

char n[50];

int s=0;

printf("\n\nEnter a string: ");

fgets(n,sizeof(n),stdin);

for(int i=0; i<strlen(n)-1;i++){

int c = n[i];

s+=c;

}

printf("\nSTRING STRENGTH CALCULATOR\n\n String: %s

\n Strength :%d", n, s);

}

**OUTPUT:**

Enter a string: Antony Xavio Immanuel

STRING STRENGTH CALCULATOR

String: Antony Xavio Immanuel

Strength :2040

## **6.** **Read a string and check if it’s a palindrome.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

char str[20], rev[20];

printf("\nPALINDROME CHECKER");

printf("\nEnter a string: ");

scanf("%s", str);

for (int i = strlen(str) - 1, j = 0; i >= 0; i--, j++)

{

rev[j] = str[i];

}

(strcasecmp(rev, str) == 0) ? printf("It is a palindrome")

: printf("It is not a palindrome");

}

**OUTPUT:**

PALINDROME CHECKER

Enter a string: Malayalam

It is a palindrome

## **7.Print the following pattern.**

**COMPUTER**

**OMPUTER**

**MPUTER**

**PUTER**

**UTER**

**TER**

**ER**

**R**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

void main()

{

char s[9] = "COMPUTER";

for (int i = 0; i < strlen(s); i++)

{

for (int j = i; j < strlen(s); j++)

{

printf("%c", s[j]);

}

printf("\n");

}

}

**OUTPUT:**

C O M P U T E R

O M P U T E R

M P U T E R

P U T E R

U T E R

T E R

E R

R

**RESULT:**

Thus all C programs related to problem solving using strings has been executed successfully.

Experiment No.: 7 Date:

# **PROGRAMS TO ILLUSTRATE STRUCTURES AND UNION**

## **1.** **Create a structure Complex and perform addition, subtraction, and multiplication of two complex numbers.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

#include <stdlib.h>

struct complex {

int real;

int img;

} c1, c2;

void main() {

printf("Enter first complex number \n");

printf("Real part: ");

scanf("%d", &c1.real);

printf("Imaginary part: ");

scanf("%d", &c1.img);

printf("Enter second complex number \n");

printf("Real part: ");

scanf("%d", &c2.real);

printf("Imaginary part: ");

scanf("%d", &c2.img);

int realadd = c1.real + c2.real;

int imgadd = c1.img + c2.img;

int realsub = c1.real - c2.real;

int imgsub = c1.img - c2.img;

int mul1 = (c1.real \* c2.real) - (c1.img \* c2.img);

int mul2 = (c1.real \* c2.img) + (c1.img \* c2.real);

printf("\nAddition: %d + %di\n", realadd, imgadd);

printf("Subtraction: %d %s %di\n",

realsub, (imgsub >= 0) ? "+" : "-", abs(imgsub));

printf("Multiplication : %d + %di\n", mul1, mul2);

}

**OUTPUT:**

Enter first complex number

Real part: 3

Imaginary part: 2

Enter second complex number

Real part: -6

Imaginary part: 1

Addition : -3 + 3i

Subtraction : 9 + 1i

Multiplication : -20 + -9i

## **2. Create a structure STUDENT with <Roll\_no, Name, Gender, marks [5], Grades [5], GPA>. Calculate the grade of each subject (assuming one-to-one mapping of marks and grades) and GPA as the average of marks scored by a student.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

typedef struct student

{

char name[30];

int roll;

char gender;

int marks[7];

float gpa;

char grade[7];

} stud;

void main()

{

char subjects[7][11] = {"Tamil", "English", "Maths", "Computer", "Physics", "Chemistry", "B.C.M"};

stud s;

printf("\n\nEnter student details and marks scored");

printf("\nName: ");

scanf(" %s", s.name);

printf("Roll: ");

scanf("%d", &s.roll);

printf("Gender (M/F): ");

scanf(" %c", &s.gender);

printf("Enter marks for %s (for 100)\n", s.name);

int sum = 0;

for (int j = 0; j < 7; j++) {

printf("\t-%s: ", subjects[j]);

scanf("%d", &s.marks[j]);

int n = s.marks[j];

s.grade[j] = (n >= 90) ? 'A' : (n >= 80) ? 'B': (n >= 70) ? 'C'

: (n >= 60) ? 'D': (n >= 50) ? 'E' : 'F';

sum += n;

}

s.gpa = sum / 7;

printf("\n\nPrinting student details and marks\n\n");

printf("\nName: %s", s.name);

printf("\nRoll number: %d", s.roll);

printf("\nGender: %c", s.gender);

printf("\nGPA: %.2f", s.gpa);

printf("\n\tMarksheet");

for (int j = 0; j < 7; j++)

printf("\n\t-%s: %d (%c)", subjects[j], s.marks[j], s.grade[j]);

}

**OUTPUT:**

Enter student details and marks scored

Name: Antony

Roll: 05

Gender (M/F): M

Enter marks for Antony (for 100)

-Tamil: 90

-English: 94

-Maths: 99

-Computer: 100

-Physics: 94

-Chemistry: 89

-E.E.E: 90

Printing student details and marks

Name: Antony

Roll number: 5

Gender: M

GPA: 93.00

Marksheet

-Tamil: 90 (A)

-English: 94 (A)

-Maths: 99 (A)

-Computer: 100 (A)

-Physics: 94 (A)

-Chemistry: 89 (B)

-E.E.E: 90 (A)

## **3. Calculate the grades and GPA of N students of a class using the STUDENT structure.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

typedef struct student{

char name[30];

int roll;

char gender;

int marks[7];

float gpa;

char grade[7];

} stud;

void main(){

char subjects[7][11]={"Tamil", "English","Maths", "Computer", "Physics", "Chemistry", "E E E"};

// getting number of students

int n;

printf("Enter the number of students: ");

scanf("%d", &n);

stud s[n];

// getting student info and marks

printf("\n\nEnter student details and marks scored");

for (int i = 0; i < n; i++){

printf("\n\n\tStudent- %d", i + 1);

printf("\nName: ");

scanf(" %s", s[i].name);

printf("Roll: ");

scanf("%d", &s[i].roll);

printf("Gender (M/F): ");

scanf(" %c", &s[i].gender);

printf("Enter marks for %s", s[i].name);

int sum = 0;

for (int j = 0; j < 7; j++){

printf("\t-%s: ", subjects[j]);

scanf("%d", &s[i].marks[j]);

int n=s[i].marks[j];

s[i].grade[j] = (n>=90)?'A':(n>=80)?'B':(n>=70)?'C':(n>=60)?'D':(n>=50)?'E':'F';

sum +=n;

}

s[i].gpa = sum / 7;

}

// Printing the details with formatting

printf("\n\nPrinting student details and marks\n\n");

for (int i = 0; i < n; i++){

printf("\nName: %s", s[i].name);

printf("\nRoll number: %d", s[i].roll);

printf("\nGender: %c", s[i].gender);

printf("\nGPA: %.2f", s[i].gpa);

printf("\n\tMarksheet");

for (int j = 0; j < 7; j++)

printf("\n\t-%s: %d (%c)", subjects[j], s[i].marks[j], s[i].grade[j]);

}

}

**OUTPUT:**

Enter the number of students: 3

Enter student details and marks scored

Student- 1

Name: Antony

Roll: 5

Gender (M/F): M

Enter marks for Antony

-Tamil: 94

-English: 86

-Maths: 99

-Computer: 100

-Physics: 93

-Chemistry: 98

-E E E: 87

Student- 2

Name: Kevin

Roll: 64

Gender (M/F): M

Enter marks for Kevin

-Tamil: 90

-English: 84

-Maths: 100

-Computer: 99

-Physics: 100

-Chemistry: 86

-E E E: 92

Student- 3

Name: Dhanesh

Roll: 35

Gender (M/F): M

Enter marks for Dhanesh

-Tamil: 100

-English: 86

-Maths: 96

-Computer: 97

-Physics: 96

-Chemistry: 99

-E E E: 100

Printing student details and marks

Name: Antony

Roll number: 5

Gender: M

GPA: 93.00

Marksheet

-Tamil: 94 (A)

-English: 86 (B)

-Maths: 99 (A)

-Computer: 100 (A)

-Physics: 93 (A)

-Chemistry: 98 (A)

-E E E: 87 (B)

Name: Kevin

Roll number: 64

Gender: M

GPA: 93.00

Marksheet

-Tamil: 90 (A)

-English: 84 (B)

-Maths: 100 (A)

-Computer: 99 (A)

-Physics: 100 (A)

-Chemistry: 86 (B)

-E E E: 92 (A)

Name: Dhanesh

Roll number: 35

Gender: M

GPA: 96.00

Marksheet

-Tamil: 100 (A)

-English: 86 (B)

-Maths: 96 (A)

-Computer: 97 (A)

-Physics: 96 (A)

-Chemistry: 99 (A)

-E E E: 100 (A)

## **4. Create a structure DOB <Day,month,year>. Include this as a sub structure in STUDENT and read and print the DOB a student along with other details**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

struct student {

char name[10];

int roll;

char gender;

struct dob {

int day;

int month;

int year;

} dob;

} s;

void main() {

printf("Enter Name: ");

scanf("%s", s.name);

printf("Enter Roll number: ");

scanf("%d", &s.roll);

printf("Enter Gender (M/F): ");

scanf(" %c", &s.gender);

printf("Enter Date Of Birth\n");

printf("Day: ");

scanf("%d", &s.dob.day);

printf("Month: ");

scanf("%d", &s.dob.month);

printf("Year: ");

scanf("%d", &s.dob.year);

printf("\n DETAILS\n");

printf("Name: %s\n", s.name);

printf("Roll number: %d\n", s.roll);

printf("Gender (M/F): %c\n", s.gender);

printf("Date Of Birth: %d-%d-%d\n", s.dob.day, s.dob.month, s.dob.year);

}

**OUTPUT:**

Enter Name: Antony

Enter Roll number: 05

Enter Gender (M/F): M

Enter Date Of Birth

Day: 24

Month: 09

Year: 2005

DETAILS

Name: Antony

Roll number: 5

Gender (M/F): M

Date Of Birth: 24-9-2005

## **5. Create a union employee <emp\_id, salary>. Read and write the employee details and demonstrate the significance of union’s space management.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

union employee {

char name[10];

int empid;

float empsalary;

} e;

void main() {

printf("Enter employee id: ");

scanf("%d", &e.empid);

printf("Enter employee salary: ");

scanf("%f", &e.empsalary);

printf("\nEmployee id: %d", e.empid);

printf("\nEmployee salary: %f", e.empsalary);

printf("\nSize of union: %lu", sizeof(union employee));

printf("\nSize of empid: %lu", sizeof(e.empid));

printf("\nSize of empsalary: %lu", sizeof(e.empsalary));

}

**OUTPUT:**

Enter employee id: 98956425341

Enter employee salary: 92500

Employee id: 1203022336

Employee salary: 92500.00

Size of union: 12

Size of empid: 4

Size of empsalary: 4

**RESULT:**

Thus all programs related to structures and union have been executed successfully.

Experiment No.: 8 Date:

# **PROGRAMS TO ILLUSTRATE POINTERS**

## **1. Implement a Pocket calculator with arithmetic operators (include increment/decrement also).**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

int a, b, \*pa, \*pb, result;

char op;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

printf("Enter operator (+, -, \*, /,increment(i),decrement(d)): ");

scanf(" pointer %c", &op);

pa = &a;

pb = &b;

switch (op) {

case '+':

result = \*pa + \*pb;

printf("Result: %d\n", result);

break;

case '-':

result = \*pa - \*pb;

printf("Result: %d\n", result);

break;

case '\*':

result = \*pa \* \*pb;

printf("Result: %d\n", result);

break;

case '/':

if (\*pb != 0)

printf("Result: %d\n",\*pa / \*pb);

else

printf("Error: Division by zero\n");

break;

case 'i':

(\*pa)++;

(\*pb)++;

printf("Incremented values: a = %d, b = %d\n", \*pa, \*pb);

break;

case 'd':

(\*pa)--;

(\*pb)--;

printf("Decremented values: a = %d, b = %d\n", \*pa, \*pb);

break;

default:

printf("Invalid operator\n");

break;

}

}

**OUTPUT:**

Enter two numbers: 99 11

Enter operator (+, -, \*, /, increment(i), decrement(d)): \*

Result: 1089

## **2. Demonstrate the effect that changing the value through the pointer to a structure variable reflects the actual value of the variable. (Hint: Initialize a structure variable; create a pointer to the variable; modify the value using pointer and show the changes).**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

struct student {

int a, b;

} point, \*p;

void main() {

point.a = 10;

point.b = 20;

printf("Before changing a=%d,b=%d", point.a, point.b);

p=&point;

p->a = 30;

p->b = 40;

printf("\nAfter changing a=%d,b=%d", point.a, point.b);

}

**OUTPUT:**

Before changing a=10,b=20

After changing a=30,b=40

## **3. Use Dynamic memory allocation to read the marks of N students in a subject and find the total, average, maximum, and minimum among them.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int\* ptr;

int n,tot=0,avg,max,min;

printf("Enter the number of subjects: ");

scanf("%d",&n);

ptr=(int\*)calloc(n,sizeof(int));

for(int i=0; i<n;i++){

printf("Enter mark for subject-%d: ",i+1);

scanf("%d",&ptr[i]);

tot+=ptr[i];

}

avg=tot/n;

max=ptr[0];

min=ptr[0];

for(int i=0;i<n;i++){

if(ptr[i]>max)

max=ptr[i];

if(ptr[i]<min)

min=ptr[i];

}

printf("\n\nTotal: %d",tot);

printf("\n\nAverage: %d",avg);

printf("\n\nMaximum: %d",max);

printf("\n\nMinimum: %d",min);

}

**OUTPUT:**

Enter the number of subjects: 7

Enter mark for subject-1: 95

Enter mark for subject-2: 89

Enter mark for subject-3: 94

Enter mark for subject-4: 100

Enter mark for subject-5: 92

Enter mark for subject-6: 87

Enter mark for subject-7: 99

Total: 656

Average: 93

Maximum: 100

Minimum: 87

## **4. Read and display the details of N books where each book has <ISBN, title, price, author>.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

struct Book{char ISBN[20];char title[100];float price;char author[100];};

void main()

{

int N;

printf("Enter the number of books (N): ");

scanf("%d", &N);

struct Book \*books = calloc(N, sizeof(struct Book));

for (int i = 0; i < N; ++i) {

printf("Enter details for Book %d:\n", i + 1);

printf("ISBN: ");

scanf("%s", (books + i)->ISBN);

printf("Title: ");

scanf("%s", (books + i)->title);

printf("Price: ");

scanf("%f", &(books + i)->price);

printf("Author: ");

scanf("%s", (books + i)->author);

}

printf("\nDetails of the books:\n");

for (int i = 0; i < N; ++i) {

printf("\nBook %d:\n", i + 1);

printf("ISBN: %s\n", (books + i)->ISBN);

printf("Title: %s\n", (books + i)->title);

printf("Price: $%.2f\n", (books + i)->price);

printf("Author: %s\n", (books + i)->author);

}

free(books);

}

**OUTPUT:**

Enter the number of books (N): 2

Enter details for Book 1:

ISBN: 978-92-95055-02-5

Title: C\_programming

Price: 998.50

Author: Dhanesh

Enter details for Book 2:

ISBN: 978-17-75093-32-9

Title: Python\_Basics

Price: 1050.00

Author: David\_Amos

Details of the books:

Book 1:

ISBN: 978-92-95055-02-5

Title: C\_programming

Price: $998.50

Author: Dhanesh

Book 2:

ISBN: 978-17-75093-32-9

Title: Python\_Basics

Price: $1050.00

Author: David\_Amos

## **5. Generate a multiplication table for user input M and N.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

#include<stdlib.h>

void main(){

int n,m;

printf("\n\nMULTIPLICATION TABLE");

printf("\nEnter the number:");

scanf("%d",&n);

printf("\nEnter the number of terms: ");

scanf("%d",&m);

int \*x=&n;

for(int\* i = &n,k=1; i<&n+m;i++,k++)

printf("\n%d x %d = %d",\*x, k,(\*x)\*(k));

}

**OUTPUT:**

MULTIPLICATION TABLE

Enter the number:137

Enter the number of terms: 5

137 x 1 = 137

137 x 2 = 274

137 x 3 = 411

137 x 4 = 548

137 x 5 = 685

**RESULT:**

Thus all programs related to pointers in C have been executed successfully.Experiment No.: 9 Date:

# **PROGRAMS TO ILLUSTRATE USER DEFINED FUNCTIONS**

## **1.** **Create functions of prototypes print\_int(int), print\_float(float), print\_char(char), print\_string(char \*). Define the function with only printf(). Read your roll\_no GPA, gender, and name and use the prototypes to print them.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void print\_int(int num)

{

printf("Roll Number: %d\n", num);

}

void print\_float(float num)

{

printf("GPA: %.2f\n", num);

}

void print\_char(char ch)

{

printf("Gender: %c\n", ch);

}

void print\_string(char \*str)

{

printf("Name: %s\n", str);

}

void main()

{

int roll\_no;

float GPA;

char gender, name[50];

printf("Enter Name: ");

scanf("%s", name);

printf("Enter Roll Number: ");

scanf("%d", &roll\_no);

printf("Enter GPA: ");

scanf("%f", &GPA);

printf("Enter Gender (M/F): ");

scanf(" %c", &gender);

printf("\n\n\tDisplaying details\n");

print\_int(roll\_no);

print\_float(GPA);

print\_char(gender);

print\_string(name);

}

**OUTPUT:**

Enter Name: Immanuel

Enter Roll Number: 05

Enter GPA: 9.4

Enter Gender (M/F): M

Displaying details

Roll Number: 5

GPA: 9.40

Gender: M

Name: Immanuel

## **2. Simulate pocket calculator operations using functions of various styles (with/without parameter/return). Pass a variable result in main () by reference to demonstrate pass by reference.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

// Functions without parameters and return values

void add(int a, int b){printf("Sum: %d\n", a + b);}

void subtract(int a, int b){printf("Difference: %d\n", a - b);}

// Functions with parameters and return values

int multiply(int a, int b){return a \* b;}

float divide(int a, int b)

{

if (b != 0){ return (float)a / b;}

else{

printf("Error: Cannot divide by zero.\n");

return 0;}

}

void main()

{

int num1, num2, result\_multiply;

char ope;

float result\_divide;

printf("Enter first number: ");

scanf("%d", &num1);

printf("Enter second number: ");

scanf("%d", &num2);

printf("Enter operation (+,-,\*,/) :");

scanf(" %c", &ope);

switch (ope)

{

case '+':

add(num1, num2);

break;

case '-':

subtract(num1, num2);

break;

case '\*':

result\_multiply = multiply(num1, num2);

printf("Product: %d\n", result\_multiply);

break;

case '/':

result\_divide = divide(num1, num2);

printf("Quotient: %.2f\n", result\_divide);

break;

default:

printf("ENTER A VALID OPERATOR(+,-,\*,/)");

}

}

**OUTPUT:**

Enter first number: 95

Enter second number: 106

Enter operation (+,-,\*,/) :\*

Product: 10070

## **3.** **Demonstrate sorting N student details using a function rank\_students (student [], n) based on their GPA. Use STUDENT <roll\_no, name, GPA>.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

#define MAX\_NAME\_LENGTH 40

void sortGPA(char name[][MAX\_NAME\_LENGTH], int roll[], float gpa[], int n)

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n - 1; j++)

{

if (gpa[j] < gpa[j + 1])

{

float tempGPA = gpa[j];

gpa[j] = gpa[j + 1];

gpa[j + 1] = tempGPA;

int tempRoll = roll[j];

roll[j] = roll[j + 1];

roll[j + 1] = tempRoll;

char tempName[MAX\_NAME\_LENGTH];

strcpy(tempName, name[j]);

strcpy(name[j], name[j + 1]);

strcpy(name[j + 1], tempName);

}

}

}

}

void main()

{

int n;

printf("Enter the number of students: ");

scanf("%d", &n);

int roll[n];

float gpa[n];

char name[n][MAX\_NAME\_LENGTH];

for (int i = 0; i < n; i++)

{

printf("Enter student\_%d name: ", i + 1);

scanf("%s", name[i]);

printf("Roll number of %s: ", name[i]);

scanf("%d", &roll[i]);

printf("GPA of %s: ", name[i]);

scanf("%f", &gpa[i]);

}

sortGPA(name, roll, gpa, n);

printf("\n%-10s%-20s%-5s\n", "ROLL NO", "Name", "GPA");

for (int i = 0; i < n; i++)

{

printf("%-10d%-20s%-5.2f\n", roll[i], name[i], gpa[i]);

}

}

**OUTPUT:**

Enter the number of students: 5

Enter student\_1 name: Antony

Roll number of Antony: 5

GPA of Antony: 9.5

Enter student\_2 name: Dhanesh

Roll number of Dhanesh: 21

GPA of Dhanesh: 9.7

Enter student\_3 name: Kevin

Roll number of Kevin: 68

GPA of Kevin: 9.7

Enter student\_4 name: Abdul

Roll number of Abdul: 8

GPA of Abdul: 9.2

Enter student\_5 name: Akash

Roll number of Akash: 14

GPA of Akash: 9.0

ROLL NO Name GPA

21 Dhanesh 9.70

68 Kevin 9.70

5 Antony 9.50

8 Abdul 9.20

14 Akash 9.00

## **4. Write programs using recursive functions for the following:**

**AIM:**

**a) Find the sum of digits of a number.**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

int rec(int n){(n == 0)?return 0:return n%10 + rec(n/10);}

void main()

{

int num;

printf("Enter a number: ");

scanf("%d", &num);

int sum = rec(num);

printf("Sum of digits: %d\n", sum);

}

**OUTPUT:**

Enter a number: 987646534

Sum of digits: 52

Enter a number: 24

Sum of digits: 6

**b) Search an element in an array (Linear search).**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

int search(int arr[], int target, int index, int size)

{

if (index == size)

return -1;

else if (arr[index] == target)

return index;

else

return search(arr, target, index + 1, size);

}

void main()

{

int i, target, size;

printf("Enter the size of the array: ");

scanf("%d", &size);

int arr[size];

printf("Enter elements of the array: ");

for (i = 0; i < size; i++)

{

scanf("%d", &arr[i]);

}

printf("Target element: ");

scanf("%d", &target);

int x = search(arr, target, 0, size);

if (x == -1)

printf("Not found");

else

printf("Index: %d", x);

}

**OUTPUT:**

Enter the size of the array: 20

Enter elements of the array: 73 5 92 44 3 65 21 89 17 42 100 12 77 55 39 99 6 28 15 81

Target element: 42

Index: 9

**c)** **Print the marks in letters (refer SVCE exam portal scenario).**

**ALGORITHM:**

**PROGRAM:**

void converter(int mark)

{

static char word[11][10] = {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};

if (mark == 0){return;}

converter(mark / 10);

int digit = mark % 10;

printf("%s", word[digit]);

}

void main()

{

int mark;

printf("Enter the exam marks:");

scanf("%d", &mark);

converter(mark);

}

**OUTPUT:**

Enter the exam marks:97

nine seven

Enter the exam marks:100

one zero zero

Experiment No.: 10 Date:

# **FILE HANDLING IN C**

## **1. Read a sentence from console and write into a file out.txt. Read out.txt, change the case of each character, and write in console.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

void main()

{

FILE \*fp1, \*fp2;

char str[50], ch;

printf("\nEnter a string: ");

fgets(str, sizeof(str), stdin);

fp1 = fopen("out.txt", "w");

if (fp1 == NULL)

{

puts("File does not exist..");

exit(1);

}

fprintf(fp1, "%s", str);

fclose(fp1);

fp2 = fopen("out.txt", "r");

(fp2 == NULL) ? exit(1) : printf("File exists\n\nString: ");

while ((ch = fgetc(fp1)) != EOF)

(isupper(ch)==1)? printf("%c",tolower(ch)) :

printf("%c",toupper(ch));

fclose(fp2);

}

**OUTPUT:**

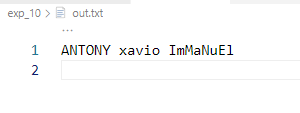
**Terminal Output:**

Enter a string: ANTONY xavio ImMaNuEl

File exists

String: antony XAVIO iMmAnUeL

**Output in out.txt file:**

****

## **2. Read any of the C files from your previous exercises and print the same on the console.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

void main()

{

char ch[50] = "";

FILE \*fp1;

fp1 = fopen("out.txt", "r");

fgets(ch, 50, fp1);

puts(ch);

fclose(fp1);

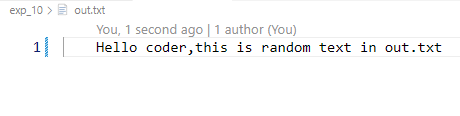
}

**OUTPUT:**

**Terminal Output:**

Hello coder, this is random text in out.txt

**Content in out.txt file:**



## **3. Read <roll, Name,GPA> of a student from console and write into a file student.txt.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

void main()

{

int n, rollno;

char name[50];

float gpa;

FILE \*fptr;

printf("Enter the number of students: ");

scanf("%d", &n);

fptr = fopen("student.txt", "w");

fprintf(fptr, "%-3s | %-10s | %-10s\n", "roll", "name", "gpa");

for (int i = 0; i < n; i++)

{

printf("Enter name: ");

scanf(" %s", name);

printf("Enter roll no.: ");

scanf(" %d", &rollno);

printf("Enter GPA: ");

scanf(" %f", &gpa);

fprintf(fptr, " %-3d | %-10s | %-10.2f\n", rollno, name, gpa);

}

fclose(fptr);

}

**OUTPUT:**

**Terminal Output:**

Enter the number of students: 3

Enter name: Antony

Enter roll no.: 5

Enter GPA: 9.3

Enter name: Dhanesh

Enter roll no.: 21

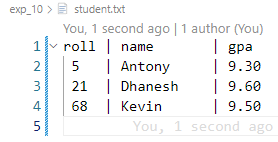
Enter GPA: 9.6

Enter name: Kevin

Enter roll no.: 68

Enter GPA: 9.5

**Output in file out.txt:**



## **4. Add a new record read from console say <100, Srinivasan,9.8> to student.txt (append).**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

void main()

{

int rollno;

char name[50];

float gpa;

FILE \*fptr;

fptr = fopen("student.txt", "a");

printf("Enter name: ");

scanf(" %s", name);

printf("Enter roll no.: ");

scanf(" %d", &rollno);

printf("Enter GPA: ");

scanf(" %f", &gpa);

fprintf(fptr, " %-3d | %-10s | %-10.2f\n", rollno, name, gpa);

fclose(fptr);

}

**OUTPUT:**

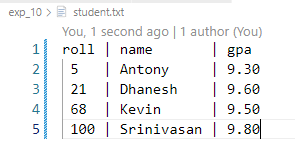
**Terminal Output:**

Enter name: Srinivasan

Enter roll no.: 100

Enter GPA: 9.8

**Output in file out.txt:**



## **5. Copy the contents of student.txt to new\_student.txt .**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

void main()

{

FILE \*fp1, \*fp2;

char s[50];

printf("\n\nCopying contents from one file to another\n");

fp1 = fopen("student.txt", "r");

fp2 = fopen("new\_student.txt", "w");

(fp1 == NULL) ? exit(1) : printf("File 1 exists\n");

(fp2 == NULL) ? exit(1) : printf("File 2 exists\n");

while(fgets(s,50,fp1)!=0)

fprintf(fp2,"%s",s);

fclose(fp2);

fclose(fp1);

}

**OUTPUT:**

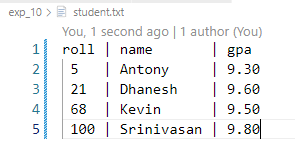
**Terminal Output:**

Copying contents from one file to another

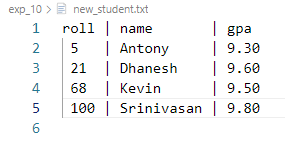
File 1 exists

File 2 exists

**Content in file student.txt:**



**Content copied to new\_student.txt:**



**RESULT:**

Thus all C programs related to file handling has been executed successfully.

Experiment No.: 11 Date:

# **ENUMERATED DATATYPES, MACROS AND STRORAGE CLASSES**

## **1. Create an enumerator for months of a year, days of a week, and display them.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

enum month{January=1, Febuary, March, April, May, June, July, August, September, October, November, December}m;

enum week{Sun=1, Mon, Tue, Wed, Thur, Fri, Sat}day;

void main()

{

m=March;

day=Wed;

printf("\n\nProgram using ENUM");

printf("\nMarch in num: %d",m);

printf("\nWed in num: %d",day);

}

**OUTPUT:**

Program using ENUM

March in num: 3

Wed in num: 4

## **2. Create an enumerator GEMS with user-defined values as price for a gram of each GEM (diamond,pearl etc). Use typedef for GEMS and print the price of a GEM.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include <stdio.h>

#include <string.h>

typedef enum GEMS { Diamond = 8000, Pearl = 20, Ruby = 300, Saphire = 450 } g;

int main(void) {

g gem=Diamond;

printf("Cost of Diamond (per gram) $%d",gem);

}

**OUTPUT:**

Cost of Diamond (per gram) $8000

## **3. Calculate the square and cube of a number using macro functions.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

# define square(a) (a\*a)

# define cube(a) (square(a)\*a)

void main() {

int n;

printf("Enter a number: ");

scanf("%d",&n);

printf("\nSquare: %d \nCube: %d ",square(n), cube(n));

}

**OUTPUT:**

Enter a number: 248

Square: 61504

Cube: 15252992

## **4. SBI generates tickets at its automated counter machine starting from 100. Each user gets a subsequent ticket number. Demonstrate this using appropriate storage class.**

**AIM:**

**ALGORITHM:**

**PROGRAM:**

#include<stdio.h>

void main()

{

printf("\n\n\tTICKET GENERATOR\n1.new user\t0.end\n\n");

int input;

while(1)

{

static int automate=100;

printf("\ninput: ");

scanf("%d",&input);

if(input==0)

break;

printf("Ticket number: %d",automate++);

}

}

**OUTPUT:**

TICKET GENERATOR

1.new user 0.end

input: 1

Ticket number: 100

input: 1

Ticket number: 101

input: 0

**RESULT:**

Thus all C programs related to Enumerated Datatype, Macros and Storage classes have been executed successfully.