**PRACTICAl -1**

**AIM:**

Write a C program that will output this passage by Michael Singer. Make sure your output looks exactly as shown here (including spacing, line breaks,punctuation, and the title and author). Use Required **Escape Sequence** and **ASCII Value.**

****

There are three shapes in the output: Smiling Face, Diamond & Heart.

The ASCII Value for Smiling face is 1.

The ASCII Value for Diamond is is 4.

The ASCII Value for Heart is is 3.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

int main()

{

char a=1,b=4,c=3;

clrscr();

printf("%c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c\n",a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a);

printf("%c",b);

printf("\"If you are resisting something,you are feeling it.\t\t\t");

printf("%c\n",b);

printf("%c",c);

printf("\tAny energy you fight, you are feeding it.\t\t\t");

printf("%c\n",c);

printf("%c",a);

printf("\t\tIf you are pushing something away,\t\t\t");

printf("%c\n",a);

printf("%c",b);

printf("\t\t\tYou are inviting it to stay.\"by Michel singer. ");

printf("%c\n",b);

printf("%c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c\n",a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a);

getch();

return 0;

}

**PRACTICAL-2**

**AIM:**

Ramesh‟s basic salary is input through the keyboard. His dearness allowance is 40% of basic salary, and house rent allowance is 20% of basic salary. Write a program to calculate his gross salary.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

int main()

{

int s,da,ha,ts;

printf("please enter ramesh's salary\n");

scanf("%d",&s);

da=(s\*40)/100;

printf("ramesh's dearness allowance is %d\n",da);

ha=(s\*20)/100;

printf("ramesh's house rant allowance is %d\n",ha);

ts=s+ha+da;

printf("ramesh total salary is %d",ts);

getch();

return 0;

}

**PRACTICAL-3**

**AIM:**

Write a program to calculate area of two circle. (πr2). Use Preprocessor directive named macro expansion for the symbol π (Symbolic Constant) without argument and with argument. Use typedef to rename the floatdatatype.

**PROGRAM CODE:**

**3 (a):**

#include<stdio.h>

#include<conio.h>

#define PI 3.14

void main()

{

typedef float circle;

circle r1,r2,a1,a2;

printf("\nEnter the value of r1:");

scanf("%f",&r1);

printf("\tEnter the value of r2:");

scanf("%f",&r2);

a1=PI\*r1\*r1;

a2=PI\*r2\*r2;

printf("\tArea of radius %f is %.2f.\n",r1,a1);

printf("\tArea of radius %f is %.2f.\n",r2,a2);

}

**3 ( b) :**

#include<stdio.h>

#include<conio.h>

#define PI 3.14

#define PERIMETER(r) 2\*PI\*r

void main()

{

typedef float circle;

circle r1,r2,a1,a2;

printf("\n\n\n\tEnter the value of r1:");

scanf("%f",&r1);

printf("\tEnter the value of r2:");

scanf("%f",&r2);

a1=PI\*r1\*r1;

a2=PI\*r2\*r2;

printf("\tArea of radius %f is %.2f.\n",r1,a1);

printf("\tArea of radius %f is %.2f.\n",r2,a2);

printf("\tPerimeter of r1 is %2f\n",PERIMETER(r1));

printf("\tPerimeter of r2 is %2f",PERIMETER(r2));

getch();

}

**PRACTICAL-4(A)**

**AIM:**

Write a program to do following:

Input an amount and convert it into rupees and paisa. (For Ex. 25.67Rs = 25 Rs and 67 Paisa).**(Implicit type Conversion)**

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

int main()

{

int R,P;

float amount;

clrscr();

printf("enter the amount\n");

scanf("%f",&amount);

R=amount;

printf("your Rupees is %d\n",R);

P=(amount-R)\*100;

printf("your paisa is %d\n",P);

getch();

return 0;

}

**PRACTICAL-4(B)**

**AIM:**

Write a program to do following:

Input No of female and No of male and calculate the ratio of femalesto males in a town. No of female and No of male are in int and ratio isin float. (For Ex No\_of\_Female = 10 & No\_of\_Male = 7 then ratio =1.43).**(Explicit type Conversion)**

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

int main()

{

int nm,nf;

float ratio;

printf("enter the number of male\n");

scanf("%d",&nm);

printf("enter the number of female\n");

scanf("%d",&nf);

ratio=(float)nf/nm;

printf("the ratio of female to male is %f",ratio);

getch();

return 0;

}

**PRACTICAL-5**

**AIM:**

While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000.If quantity and price per item are input through the keyboard, write a program to calculate the total expenses. Use **Simple If** statement.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

int main()

{

int n,ppr;

float expenses;

printf("please enter a number of item you perchased\n");

scanf("%d",&n);

printf("please enter price per item\n");

scanf("%d",&ppr);

if(n>1000)

{

expenses=0.9\*ppr\*n;

printf("your total expenses is %f\n",expenses);

}

else

{

expenses=(ppr\*n);

printf("your total expenses is %f\n",expenses);

}

getch();

return 0;

}

**PRACTICAL-6**

**AIM:**

Given three points (x1, y1), (x2, y2) and (x3, y3), write a program to check if all the three points fall on one straight line. Use fabs() function of <maths.h>header file. Use if…elsestatement

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int x1,x2,x3,y1,y2,y3;

float s1,s2:

printf("\nEnter x1 ");

scanf("%d",&x1);

printf("Enter y1 ");

scanf("%d",&y1);

printf("Enter x2 ");

scanf("%d",&x2);

printf("Enter y2 ");

scanf("%d",&y2);

printf("Enter x3 ");

scanf("%d",&x3);

printf("Enter y3 ");

scanf("%d",&y3);

s1=fabs((y2-y1)/(x2-x1));

s2=fabs((y3-y2)/(x3-x2));

if(s1==s2)

{

printf("The points are colinear.");

}

else

{

printf("The points non colinear .");

}

getch();

}

**PRACTICAL-7**

**AIM:**

If the three sides of a triangle are entered through the keyboard, write a program to check whether the triangle is valid or not. The triangle is valid if the sum of two sides is greater than the largest of the three sides. Use nested if…else statement.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

int s1,s2,s3;

printf("\nEnter three sides of triangle");

scanf("\n%d %d %d",&s1,&s2,&s3);

if((s1+s2)>s3)

{

if((s2+s3>s1))

{

if((s3+s1)>s2)

{

printf("The triangle is valid.");

}

}

}

else

{

printf("The triangle is not valid.");

}

getch();

}

**PRACTICAL-8**

**AIM:**

An Insurance company follows following rules to calculate premium.

(1) If a person‟s health is excellent and the person is between 25 and 35 years of age and lives in a city and is a male then the premium is Rs. 4 per thousand and his policy amount cannot exceed Rs. 2 lakhs.

(2) If a person satisfies all the above conditions except that the gender is female then the premium is Rs. 3 per thousand and her policy amount cannot exceed Rs. 1 lakh.

(3) If a person‟s health is poor and the person is between 25 and 35 years of age and lives in a village and is a male then the premium is Rs. 6 per thousand and his policy cannot exceed Rs. 10,000.

(4) In all other cases the person is not insured. Write a program to output whether the person should be insured or not, his/her premium rate and maximum amount for which he/she can be

insured. Use Else…if Ladder.

**PROGRAM-CODE:**

#include<stdio.h>

#include<conio.h>

int main()

{

char health,place,gender;

int age;

printf("please enter your health like for excelant press e and for poor press p\n");

scanf("%c",&health);

fflush(stdin);

printf("please enter your place like for city press c and for village press v\n");

scanf("%c",&place);

fflush(stdin);

printf("please enter your gender like for male press m and for female press f\n");

scanf("%c",&gender);

fflush(stdin);

printf("please enter your age\n");

scanf("%d",&age);

if((health=='e'||health=='E')&&(place=='c'||place=='C')&&(gender=='m'||gender=='M')&&(age>=25)&&(age<=35))

{

printf("the primium is rs 4 per thousand and his policy amount can not exceed rs 2 lakh\n");

}

if((health=='e'||health=='E')&&(place=='c'||place=='C')&&(gender=='f'||gender=='F')&&(age>=25)&&(age<=35))

{

printf("the primium is rs 3 per thousand and his policy amount can not exceed rs 1 lakh\n");

}

if((health=='p'||health=='P')&&(place=='v'||place=='V')&&(gender=='m'||gender=='M')&&(age>=25)&&(age<=35))

{

printf("the primium is rs 6 per thousand and his policy amount can not exceed rs 10000\n");

}

else

{

printf("u r not valid for policy\n");

}

getch();

return 0;

}

**PRACTICAL-9**

**AIM:**

Write a program to input a character using **getchar()** and print the character using **putchar()** and check the character category. Also convert uppercase alphabet to lower case and vice versa. (Use **Character Test Functions** : isalnum(), isalpha(), isdigit(), islower(), isprint(), ispunct(), isspace(), isupper()) and (toupper() & tolower()) of <ctype.h> header file.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

void main()

{

char a;

printf("\nEnter your character ");

a=getchar();

if((isalnum(a))&&(isprint(a)))

{

printf("\nThe input is printable\n");

if(isalpha(a))

{

printf("\nThe character is an alphabet\n");

if(isupper(a))

{

printf("\nThe character is in upper case\n");

a=tolower(a);

putchar(a);

}

else if(islower(a))

{

printf("\nThe character is in lower case\n");

a=toupper(a);

putchar(a);

}

}

if(isdigit(a))

{

printf("\nThe character is digit.");

putchar(a);

}

}

else if((ispunct(a))&&(isprint(a)))

{

printf("\nThe input is a punctuation.");

putchar(a);

}

else if((isspace(a))&&(isprint(a)))

{

printf("\nThe input is a blank space and not printable.");

putchar(a);

}

getch();

}

**PRACTICAL-10**

**AIM:**

Write a program to find the grace marks for a student using **Switch Statement**. The user should enter the class obtained by the student and the number of subjects he has failed in.

1. If the student gets first class and the number of subjects he failed in is greater than 3, then he does not get any grace. If the number of subjects he failed in is less than or equal to 3 then the grace is of 5 marks per subject.

2. If the student gets second class and the number of subjects he failed in is greater than 2, then he does not get any grace. If the number of subjects he failed in is less than or equal to 2 then the grace is of 4 marks per subject.

3. If the student gets third class and the number of subjects he failed in is greater than 1, then he does not get any grace. If the number of subjects he failed in is equal to 1 then the grace is of 5 marks per subject.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

Int fail\_sub;

char Class;

clrscr();

printf("\nEnter class : ");

scanf("%c",&Class);

printf("\nEnter number of subjects stdent is failing : ");

scanf("%d",&fail\_sub);

switch(Class)

{

case 'f': if(fail\_sub>3)

{

printf("\nNo grace marks given");

}

else if(fail\_sub<=3)

{

printf("\n5 grace marks per subject are given");

}

break;

case 's': if(fail\_sub>2)

{

printf("\nNo grace marks given.");

}

else if(fail\_sub<=2)

{

printf("\n4 grace marks per subject are given.");

}

break;

case 't': if(fail\_sub>1)

{

printf("\nNo grace marks given.");

}

else if(fail\_sub>=1)

{

printf("5 grace marks per subject are given.");

}

break;

default: printf("Please enter correctly.");

}

getch();

}

**PRACTICAL-11**

**AIM:**

Write a program to calculate following series using if and goto statement. Compare the results using for loop.

+ +…+

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

intn,a=1,i;

longint sum1=0,sum2=0;

printf("\nEnter number of terms.");

scanf("%d",&n);

flag:

sum1=sum1+(a\*a);

a++;

if(a<=n)

goto flag;

printf("\nThe sum(using goto label) is : %d ",sum1);

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

{

sum2=sum2+(i\*i);

}

printf("\nThe sum(using for loop) is : %d",sum2);

getch();

}

**PRACTICAL-12**

**AIM:**

An “Armstrong number” is an n-digit number that is equal to the sum of the nth powers of its individual digits. For example, 153 is an Armstrong number because it has 3 digits and 13+ 53+33 =153. Similarly 1634 is an Armstrong number because it has 4 digits and 14+ 64+34 + 44 = 1634. Write a program to find whether the entered number is Armstrong or not using While Loop.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

int main()

{

int sum=0,on,num,c,n,p;

printf("enter a num. :");

scanf("%d",&on);

num=on;

n=0;

while(on>0)

{

on=on/10;

n++;

}

on=num;

while(on>0)

{

c=on%10;

p=pow(c,n);

sum=sum+p;

on=on/10;

}

if(num==sum)

{

printf("number is armstrong");

}

else

{

printf("number is not armstrong");

}

getch();

return 0;

}

**PRACTICAL-13**

**AIM:**

Write a menu driven program which has following options:

1. Prime or not

2. Perfect number or not

3. Factorial of a number

4. Exit

Use do...while statement so that the menu is displayed at least once. Also use switch statement.

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

Int choice,num;

while(1)

{

printf("\nMENU!");

printf("\n1.To check whether number is prime or not.");

printf("\n2.To check wheteher number is perfect or not.");

printf("\n3.To calculate factorial of the number.");

printf("\n4.To exit the program.");

printf("\nEnter your choice.");

scanf("%d",&choice);

switch(choice)

{

case 1: {

int c=0,i;

printf("\n\nPrime Number or not");

printf("\nEnter your number.");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

if(num%i==0)

c++;

}

if(c==2)

printf("\nNumber is prime.");

else

printf("\nNot prime.");

getch();

clrscr();

}

break;

case 2: {

inti=0,sum=0;

printf("\n\nPerfectNumberor not");

printf("\nEnter your number.");

scanf("%d",&num);

for(i=1;i<num;i++)

{

if(num%i==0)

sum=sum+i;

}

if(sum==num)

printf("\nThe number is perfect number.");

getch();

clrscr();

}

break;

case 3: {

int fact=1,i;

printf("\n\nFactorial of number");

printf("\nEnter your number.");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

fact=fact\*i;

}

printf("\nThe factorial is : %d ",fact);

getch();

clrscr();

}

break;

case 4: {

printf("Thank you!Exiting the program(Press any key to exit)");

getch();

exit(0);

}

break;

default: printf("\nNot valid choice.");

getch();

clrscr();

}

}

}

**PRACTICAL-14(A)**

**AIM:**

Write a program to print the following pattern using Nested for loop. (Any one in homework)

a) \*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int n=5,i,j;

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

{

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

{

printf("\* ");

}

printf("\n");

}

}

**PRACTICAL-14(B)**

**AIM:**

Write a program to print the following pattern using Nested for loop. (Any one in homework)

b) 1

3 5

7 9 11

13 15 17 19

21 23 25 27 29

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int n=5,i,j,p=1;

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

{

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

{

printf("%d ",p);

p=p+2;

}

printf("\n");

}

}

**PRACTICAL-14(C)**

**AIM:**

Write a program to print the following pattern using Nested for loop. (Any one in homework)

c)

1

1 2

3 5 8

13 21 34 55

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

Int i,j,k,a=0,b=1,c;

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

{

for(j=4;j>=i;j--)

{

printf(" ");

}

for(k=1;k<=i;k++)

{

c=a+b;

printf("%3d ",c);

b=a;

a=c;

}

printf("\n");

}

getch();

}

**PRACTICAL-14(D)**

**AIM:**

Write a program to print the following pattern using Nested for loop.

(Any one in homework)

d) A A A A A

B B B B

C C C

D D

F

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int i,j,n=5,k,a=65;

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

{

for(k=2;k<=i;k++)

{

printf(" ");

}

for(j=5;j>=i;j--)

{

printf("%c ",a);

}

printf("\n");

a++;

}

}

**PRACTICAL-H1**

**AIM:**

Write a program for a match-stick game between the computer and a user. Your Program should ensure that the computer always wins. Rules for the games are as follows:

* There are 21 match-sticks.
* The computer asks the player to pick 1, 2, 3, or 4 match-sticks.
* After the person picks, the computer does its picking.
* Whoever is forced to pick up the last match-stick loses the game.

**Use while loop, break and Continue Statements.**

To understand the above game in a better way visit the following link:  
http://atozmath.com/Games/21MatchStick.aspx

**PROGRAM CODE:**

#include<stdio.h>;

int main()

{

int i,strick=21,num,comp;

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

{

printf("|");

}

while(strick!=1)

{

printf("\n enter a chose=");

scanf("%d",&num);

while(num!=1 && num!=2 && num!=3 && num!=4)

{

printf("invalid number");

scanf("%d",&num);

}

comp=5-num;

strick=strick-5;

printf(" computer choose %d sticks\n",comp);

}

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

{

printf(" remaining stick |");

printf("\n");

}printf("\n its your turn you lose the game");

}

**PRACTICAL-15**

**AIM:**

Write a program to that will scan an array from the user and print maximum and minimum number on the screen. Also find the index of given specific element from the user.

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int a[50],n,i,p,max,min;

printf("enter a number: ");

scanf("%d",&n);

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

{

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

}

max=a[0];

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

{

if(a[i]>max)

{

max=a[i];

p=i;

}

}

printf("a[%d] is max and value is %d",p,max);

min=a[0];

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

{

if(a[i]<min)

{

min=a[i];

p=i;

}

}

printf("\na[%d] is min and value is %d",p,min);

}

**PRACTICAL-16**

**AIM:**

Write a program to sort elements of an array in ascending order using Bubble Sort.

****

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int a[50],n,i,s,k;

printf("enter a number: ");

scanf("%d",&n);

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

{

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

}

for(k=0;k<n;k++)

{

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

{

if(a[i]>a[i+1])

{

s=a[i];

a[i]=a[i+1];

a[i+1]=s;

}

}

}

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

{

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

}

}

**PRACTICAL:17**

**AIM: Write a program that reads an M x N matrix A and prints its elements in spiral order. You should start from the element in the 0th row and 0th column in the matrix and proceed in a spiral order as shown below.**

**Output for the above matrix:1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int m=3,n=4,l=0,i,j,k=0,a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}}; while(k<m && l<n)

{

for(i=l;i<n;i++)

{

printf("%d ",a[k][i]);

}

k++;

for(i=k;i<m;i++)

{

printf("%d ",a[i][n-1]);

}

n--;

if(k<m)

{

for(i=n-1;i>=l;i--)

{

printf("%d ",a[m-1][i]);

}

m--;

}

if(l<n)

{

for(i=m-1;i>=k;i--)

{

printf("%d ",a[i][l]);

}

l++;

}

}

return 0;

}

**PRACTICAL:18**

**AIM:**

**Write a program to find whether two given strings are permutations of each other. A string str1 is a permutation of str2 if all the characters in str1 appear the same number of times in str2 and str2 is of the same length as str1. For example if two strings are kindness and ksnisden the answer is Yes.**

**PROGRAM CODE:**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char s1[20],s2[20];**

**int x,l1,l2,i,j,temp,flag,n;**

**printf("enter string s1:");**

**gets(s1);**

**printf("\n enter string s2:");**

**gets(s2);**

**l1=strlen(s1);**

**l2=strlen(s2);**

**n=l1;**

**if(l1==l2)**

**{**

**temp=0;**

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

**{**

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

**{**

**if(s1[j]>s1[j+1])**

**{**

**temp=s1[j];**

**s1[j]=s1[j+1];**

**s1[j+1]=temp;**

**}**

**}**

**}**

**flag=0;**

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

**{**

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

**{**

**if(s2[j]>s2[j+1])**

**{**

**flag=s2[j];**

**s2[j]=s2[j+1];**

**s2[j+1]=flag;**

**}**

**}**

**}**

**x=strcmp(s1,s2);**

**if(x==0)**

**{**

**printf("\n yes s1 is a permutation of s2");**

**}**

**else**

**{**

**printf("\n no s1 is not a permutation of s2");**

**}**

**}**

**else**

**{**

**printf("\n string length is not same");**

**}**

**return 0;**

**}**

**PRACTICAL:H2**

**AIM:**

**Write a program that would sort a list of names of fruits in alphabetical order. For example 5 names Orange, Pineapple, Grapes, Banana, Mango should sort the names as follows: Banana Grapes Mango Orange Pineapple**

**PROGRAM CODE:**

#include<stdio.h>

#include<string.h>

int main()

{

char f[5][10]={"banana","graps","mango","apple","pineapple"};

char temp[10];

int round,i,r;

for(i=0;i<5;i++)

{

puts(f[i]);

}

printf("\n sorted order of the list is:\n");

for(round=0;round<=4;round++)

{

for(i=0;i<4-round;i++)

{

r=strcmp(f[i],f[i+1]);

if(r>0)

{

strcpy(temp,f[i]);

strcpy(f[i],f[i+1]);

strcpy(f[i+1],temp);

}

}

}

for(i=0;i<5;i++)

{

puts(f[i]);

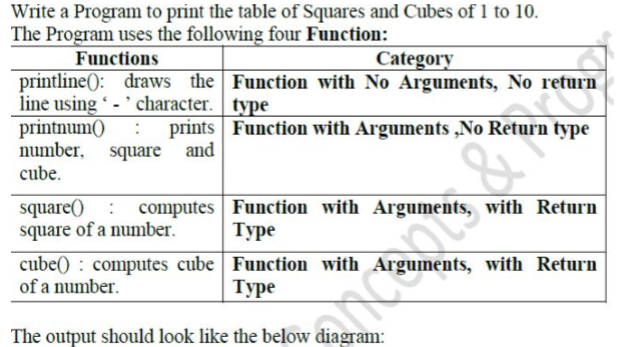
}

return 0;

}

**PRACTICAL:19**

AIM:



**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

int square(int i);

int cube(int i);

void printline();

void printnum();

int main()

{

printline();

int j;

printf(":Number : Square : cube :\n");

printline();

for(j=0;j<=10;j++)

{

printnum(j);

}

printline();

return 0;

}

int square(int i)

{

int a;

a=i\*i;

return(a);

} int cube(int i)

{ int m;

m=i\*i\*i;

return(m);

} void printline()

{

printf("---------------------------\n");

}

void printnum(int j)

{

int c=cube(j);

int b=square(j);

printf(": %6d : %6d : %6d :",j,b,c);

printf("\n");

}

**PRACTICAL-H3**

**AIM:**

Write a program to calculate nCr using Function with No arguments But with Return type.

(Hint: nCr = n! / ((r!) (n – r)!)).

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void nCr();

void main()

{

int f;

nCr();

printf("\n\nwritten by dhruv kachhadiya");

getch();

}

void nCr()

{

int n,r,i,f1=1,f2=1,f3=1,ans;

printf("enter values of n :");

scanf("%d",&n);

printf("enter value of r:");

scanf("%d",&r);

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

f1=f1\*i;

for(i=1;i<=r;++i)

f2=f2\*i;

for(i=1;i<=(n-r);++i)

f3=f3\*i;

ans = f1 / ( f2 \* f3);

printf("the nCr is = %d",ans);

}

**PRACTICAL:20**

**AIM:**

**Write a program to pass a number entered through keyboard as an argument to user-defined functions and find the factors of a number and check whether the factors are prime or not using Nested Functions PROGRAM CODE:**

#include<stdio.h>

void factor(int n);

void prime(int x);

int main()

{

int n;

printf("enter the number greater than 2:");

scanf("%d",&n);

printf("\n 1 is factor \n 1 is nutral ");

factor(n);

printf("\n %d is factor",n);

if(n%2==0)

{

printf("\n %d is not a prime number");

}

else

{

printf("\n %d is prime number");

}

return 0;

}

void factor(int n)

{

int i;

for(i=2;i<n;i++)

{

if(n%i==0)

{

printf("\n %d is factor",i);

prime(i);

}

}

}

void prime(int x)

{

int j,count=0;

for(j=2;j<x;j++)

{

if(x%j==0)

{

count++;

}

}

if(count!=0)

{

printf("\n %d is not a prime number",x);

}

else

{

printf("\n %d is a prime number",x);

}

}

**PRACTICAL:21**

**AIM:**

**Write a program to generate Fibonacci series using Recursive Function. In a Fibonacci sequence the sum of two successive terms gives the third term. 1 1 2 3 5 8 13 ….**

**PROGRAM CODE:**

#include<stdio.h>

int fib(int n);

int main()

{

int n,i;

printf("enter the number of terms:");

scanf("%d",&n);

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

{

printf("%d \n",fib(i));

}

return 0;

}

int fib(int n)

{

if(n==0||n==1)

{

return 1;

}

else

{

return(fib(n-1)+fib(n-2));

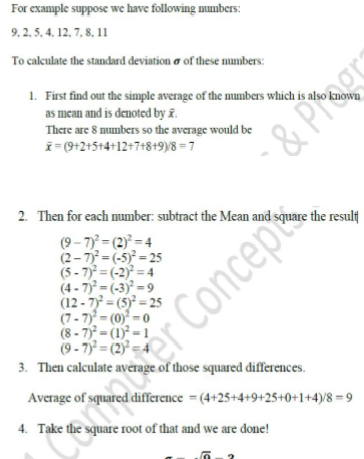
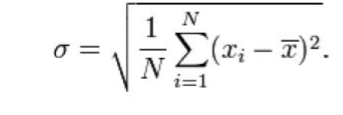
}

}

**​​PRACTICAL:22**

**AIM:**

Write a Program to compute the standard Deviation of N Numbers using Arrays & Function**.**



**PROGRAM CODE:**

#include<stdio.h>

int n;

float avg(int c[]);

int main()

{

int a[50],i,b[50],sigma;

float mean,meansquare;

printf("enter the number of element n:");

scanf("%d",&n);

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

{

printf("\n enter your element=");

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

}

mean=avg(a);

printf("\n mean=%f",mean);

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

{

b[i]=(a[i]-mean)\*(a[i]-mean);

printf("\n b[%d] =%d",i,b[i]);

}

meansquare=avg(b);

printf("\n meansquare=%f",meansquare);

sigma=sqrt(meansquare);

printf("\n sigma=%d",sigma);

return 0;

}

float avg(int c[])

{

int i,sum=0;

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

{

sum=sum+c[i];

}

return(sum/n);

}

**PRACTICAL-H4**

**AIM:**

**Write a Program to reverse a string using Recursive Function and check whether it is palindrome or not.**

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void rev()

{

int x;

if((x=getchar())!='\n')

{

rev();

printf("%c",x);

}

}

void main()

{

rev();

getch();

}

**PRACTICAL-23**

**AIM:**

**Write a Program to find the upper triangle in the given matrix. Consider the following 4 x 4 Matrix.**

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **X** | **X** | **X** |
| **0** | **X** | **X** | **X** |
| **0** | **0** | **X** | **X** |
| **0** | **0** | **0** | **X** |

**If all the elements denoted by X are non-zero then the matrix has upper triangle. For the upper triangle, all the elements of principle diagonal and above.**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int flag=1,sum=0,matrix[4][4],i,j,k;

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

{

for(j=0;j<4;j++)

{

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

}

}

printf("\nMATRIX: \n");

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

{

printf("\t\t");

for(j=0;j<4;j++)

{

printf("%2d ",matrix[i][j]);

}

printf("\n");

}

for(k=0;k<4;k++)

{

for(i=k,j=0;j<k;j++)

{

if(matrix[i][j]!=0)

{

flag=0;

break;

}

}

if(flag==0)

break;

}

if(flag==1)

printf("upper triangular\n");

if(flag==0)

printf("not upper triangular\n");

for(k=0;k<4;k++)

{

for(i=k,j=k;j<4;j++)

{

sum=sum+matrix[i][j];

}

}

printf("\nsum of upper triangle elements : %d",sum);

return 0;

}

**PRACTICAL-H5**

**AIM:**

**Write four small programs to illustrate the use of 4 storage class specifier’s auto, static, register and extern.**

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void auto\_name()

{

int x=0;

x++;

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

}

void static\_name()

{

static int w=0;

w++;

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

}

void extern\_name()

{

int y=1;

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

}

void extern\_name2()

{

int y=2;

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

y++;

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

}

void register\_name()

{

register int z=1;

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

z++;

}

void main()

{

int i;

extern int y;

printf("Output of Auto variable : \n");

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

auto\_name();

printf("\n\nOutput of Static variable : \n");

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

static\_name();

printf("\n\nOutput of Extern variable : \n");

extern\_name();

extern\_name2();

printf("\n\nOutput of Register variable : \n");

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

register\_name();

getch();

}

**PRACTICAL-24**

**AIM:**

**Create a Structure called library to hold accession number, title of the book ,author name, price of the book and flag indicating whether the book is issued or not.(flag = 1 if the book is issued , flag = 0 otherwise). Write a program to enter data of one book and display the data. Write this same program with Union also.**

**PROGRAM CODE:**

#include<stdio.h>

struct library

{

int acc\_no;

char booktitel[100];

float price;

char auther[100];

int flag;

};

int main()

{

struct library book1;

printf("enter acc no.\n");

scanf("%d",&book1.acc\_no);

fflush(stdin);

printf("enter book titel\n");

gets(book1.booktitel);

fflush(stdin);

printf("enter book price\n");

scanf("%f",&book1.price);

fflush(stdin);

printf("enter book auther\n");

gets(book1.auther);

fflush(stdin);

printf("\nplease enter 1 for book isuued and 0 for book not issued\n");

fflush(stdin);

scanf("%d",&book1.flag);

fflush(stdin);

printf("%d\n",book1.acc\_no);

printf("%s\n",book1.booktitel);

printf("%f\n",book1.price);

printf("%s\n",book1.auther);

if(book1.flag==1)

{

printf("book is issued");

}

else

{

printf("book is not issued");

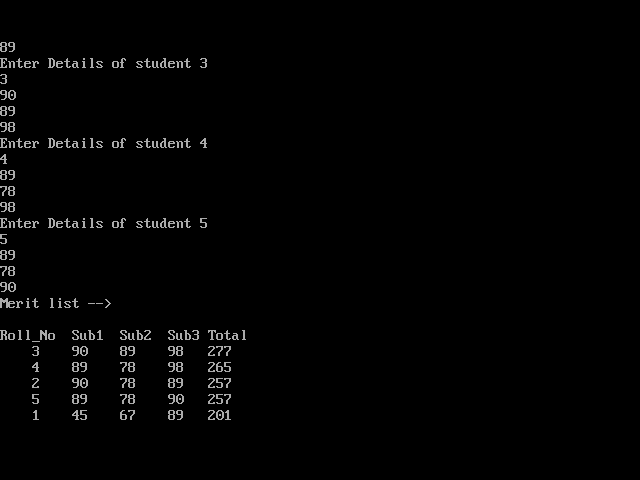
}

}

**PRACTICAL-25**

**AIM:**

**Define a structure called Result for students. Structure will have members like Roll number, marks for three subjects and total of three subjects. Write a program to enter data for 5 students and display the merit list of students. Use Array of Structures. For example, if Roll No and marks of three subjects of each student are entered through the keyboard , the output should look like the following:**



**PROGRAM CODE:**

#include<stdio.h>

struct result

{

int rno;

int m1,m2,m3,total,i,j;

};

int main()

{

struct result r[50],temp;

int i,j;

for(i=0;i<5;i++)

{

printf("roll no");

scanf("%d",&r[i].rno);

printf("m1");

scanf("%d",&r[i].m1);

printf("m2");

scanf("%d",&r[i].m2);

printf("m3");

scanf("%d",&r[i].m3);

r[i].total=r[i].m1+r[i].m2+r[i].m3;

}

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

{

for(j=0;j<4;j++)

{

if(r[j].total<r[j+1].total)

{

temp=r[j];

r[j]=r[j+1];

r[j+1]=temp;

}

}

}

printf("roll-no sub1 sub2 sub3 total \n");

for(i=0;i<5;i++)

{

printf(" %d\t",r[i].rno);

printf(" %d\t",r[i].m1);

printf("%d\t",r[i].m2);

printf("%d\t",r[i].m3);

printf("%d\t",r[i].total);

printf("\n");

}

}

**PRACTICAL-26**

**AIM:**

**Write a program to read and display information of salary of an employee using Structure within a Structure. Outer structure contains members like name of employee, designation, department name, basic pay and inner structure contains dearness allowance, house\_rent allowance and city\_allowance. Calculate the total salary of one employee.**

**PROGRAM CODE:**

#include<stdio.h>

struct pd

{

char n[100],des[50],dep[100];

float bp;

struct ea

{

float da,hra,ca,gs;

}alw;

};

int main()

{

struct pd e;

printf("Enter name:");

gets(e.n);

printf("\nEnter designation:");

gets(e.des);

printf("\nEnter department:");

gets(e.dep);

printf("\nBasic Pay:");

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

printf("\nDearness Allowance:");

scanf("%f",&e.alw.da);

printf("\nHouse Rent Allowance:");

scanf("%f",&e.alw.hra);

printf("\nCity Allowance:");

scanf("%f",&e.alw.ca);

e.alw.gs=e.alw.ca+e.alw.da+e.alw.hra+e.bp;

printf("\nGross Salary:%0.2f Rs.",e.alw.gs);

return 0;

}

**PRACTICAL-27**

**AIM:**

**Define a structure named Date that contains three members day, month and Year. Write a program that compares two given dates. If the dates are equal then display message as “Equal” otherwise “Unequal”. Write a function Check\_Date to check whether the entered date is proper or not. The date is proper if day is between 1 and 31, month is between 1 and 12 and year is between 1000 and 9999. (Structures & Functions)**

**PROGRAM CODE:**

#include<stdio.h>

struct date

{

int day;

int month;

int year;

}d1,d2;

int check\_date(struct date d);

int main()

{

int x,y;

printf("enter day of d1 : ");

scanf("%d",&d1.day);

printf("enter month of d1 : ");

scanf("%d",&d1.month);

printf("enter year of d1 : ");

scanf("%d",&d1.year);

printf("enter day of d2 : ");

scanf("%d",&d2.day);

printf("enter month of d2 : ");

scanf("%d",&d2.month);

printf("enter year of d2 : ");

scanf("%d",&d2.year);

x=check\_date(d1);

y=check\_date(d2);

if(x==1 && y==1)

{

if(d1.day==d2.day && d1.month==d2.month && d1.year==d2.year)

{

printf("Equal");

}

else

{

printf("not Equal");

}

}

else

{

printf("date format is wrong....comparision is not possible");

}

}

int check\_date(struct date d)

{

if(d.day<=31 && d.day>=1)

{

if(d.month>=1 && d.month<=12)

{

if(d.year>=1000 && d.year<=9999)

{

return 1;

}

}

}

else

{

return 0;

}

}

**PRACTICAL-28**

**AIM:**

**Write a program to perform following operations on two integer pointers.**

1. **Addition**
2. **Subtraction**
3. **Increment**
4. **Swaping of two numbers**

**Max and min of two numbers**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int a,b;

int \*p;

int \*q;

int swap;

printf("ENTER VALUE OF a : ");

scanf("%d",&a);

printf("ENTER VALUE OF b : ");

scanf("%d",&b);

p=&a;

q=&b;

printf("a=%d b=%d \n\n\n",a,b);

printf("addition : \n");

printf("%d\n",(\*p + \*q));

printf("substraction : \n");

printf("%d\n",(\*p - \*q));

printf("increment : \n");

printf("%d %d\n",++\*p,++\*q);

printf("swaping : \n");

swap=\*p;

\*p=\*q;

\*q=swap;

printf("%d %d \n\n",\*p,\*q);

if(\*p>\*q)

{

printf("%d is max and %d is min",\*p,\*q);

}

else if(\*p==\*q)

{

printf("both are equal");

}

else

{

printf("%d is max and %d is min",\*q,\*p);

}

}

**PRACTICAL-29**

**AIM:**

**Write a program to read the marks of 10 students for the subject CE141 Computer concepts and Programming and computes the number of students in categories FAIL, PASS, FIRST CLASS and DISTINCTION using Pointers and Arrays.**

|  |  |
| --- | --- |
| **Marks** | **Categories** |
| **70 or Above** | **DISTINCTION** |
| **69 to 60** | **FIRST CLASS** |
| **59 to 40** | **PASS** |
| **Below 40** | **FAIL** |

**For example if following marks of 10 students are entered:**

**34 56 78 98 12 31 67 75 91 23**

**Then the output should be**

**DISTINCTION 4 FIRST CLASS 1 PASS 1 FAIL 4**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int a[50];

int n,i;

int dis=0,first=0,pass=0,fail=0;

printf("enter num. of student ");

scanf("%d",&n);

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

{

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

}

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

{

if(\*(a+i)>=70)

{

dis++;

}

else if(\*(a+i)>=60 && \*(a+i)<=69)

{

first++;

}

else if(\*(a+i)>=40 && \*(a+i)<=59)

{

pass++;

}

else

{

fail++;

}

}

printf("DISTINCTION=%d FIRST\_CLASS=%d PASS=%d FAIL=%d ",dis,first,pass,fail);

}

**PRACTICAL-H6**

**AIM:**

**Write a program that extracts part of the given string from the specified position. For example, if the string is “Workshop on Cloud Computing”, then if from position 5, 4 characters are to be extracted then the program should return the string as “shop”. Moreover, if the position from where the string is to be extracted is given and the number of characters to be extracted is 0 then the program should extract entire string from the specified position. (Pointers and Strings)**

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

void main()

{

char s[100];

int i=0,n,pos;

clrscr();

printf("enter the string : ");

gets(s);

printf("\nenter the position to extract from: ");

scanf("%d",&pos);

printf("\nenter the number of characters to extract: ");

scanf("%d",&n);

printf("\nExtracted string : ");

if(n==0)

{

while(s[i]!='\0')

{

if(i>=pos-1)

{

putch(s[i]);

}

i++;

}

}

else

{

while(s[i]!='\0')

{

if(i>=pos-1 && i<=pos-1+(n-1))

{

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

}

i++;

}

}

getch();

}

**PRACTICAL-30**

**AIM:**

**Write a program that uses an array of pointers to strings str[ ]. Receive two strings str1 and str2 and check if str1 is embedded in any of the strings in str[ ]. If str1 is found, then replace it with str2.**

**char \*str[ ] = {**

**"We will teach you how to...",**

**"Move a mountain",**

**"Level a building",**

**"Erase the past",**

**"Make a million",**

**"...all through C!"**

**} ;**

**For example if str1 contains "mountain" and str2 contains "car", then the second string in str should get changed to "Move a car".**

**(Array of Pointers)**

**PROGRAM CODE:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void replace();

int main()

{

char \*str[] = {

"We will teach you how to...",

"Move a mountain",

"Level a building",

"Erase the past",

"Make a million",

"...all through C !"

};

char str1[80],str2[80];

int i;

printf("\n\n");

for(i=0;i<6;i++)

{

printf("\t%s\n",\*(str+i));

}

printf("\n\n");

printf("Enter the word to search: ");

gets(str1);

printf("\n\nEnter the word to replace: ");

gets(str2);

printf("\nBefore modification:\n\n");

for(i=0;i<6;i++)

{

printf("\t%s\n",\*(str+i));

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* passing all strings to replace function \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

printf("\nAfter modification:\n\n");

for(i=0;i<6;i++)

{

replace(\*(str+i),str1,str2);

}

return 0;

}

void replace(char \*s, char s1[80], char s2[80]) {

int i=0,j=0,k=0;

char temp[100],temp2[100],main[100],\*t=temp;

/\* copying to temporary string \*/

while(\*s!='\0') {

\*t=\*s;

t++;

s++;

}

\*t='\0';

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* checking each word \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(temp[i]!='\0') {

temp2[j]=temp[i];

if(temp[i]==' ') {

temp2[j]='\0';

if(strcmpi(temp2,s1)==0) {

strcpy(temp2,s2);

}

j=0;

while(temp2[j]!='\0') {

main[k]=temp2[j];

k++;

j++;

}

main[k]=' '; /\* adding space after each word is copied \*/

k++; /\* increment so that the next word won't replace the space \*/

j=-1;

}

i++;

j++;

}

temp2[j]='\0'; /\* last word terminated \*/

if(strcmpi(temp2,s1)==0){ /\* checking last word too \*/

strcpy(temp2,s2);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* last word of the string \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

j=0;

while(temp2[j]!='\0') {

main[k]=temp2[j];

k++;

j++;

}

main[k]='\0'; /\* new string is completely ready \*/

printf("\t%s\n",main); /\* printing the new string \*/

}

**PRACTICAL-31**

**AIM:**

**Write a program which performs the following tasks:**

**− initialize an integer array of 10 elements in main( )**

**− pass the entire array to a function modify( )**

**− in modify( ) multiply each element of array by 3**

**− return the control to main( ) and print the new array elements in**

**main( )**

**Above program should be done in two ways: call by value and call by address and illustrate the difference between them. (Pointers as Function Arguments)**

**PROGRAM CODE:**

**1.**

#include<stdio.h>

#include<conio.h>

void modify(int\*);

void main()

{

int a[10],i;

clrscr();

for(i=0;i<10;i++)

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

modify(a);

for(i=0;i<10;i++)

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

printf("\n");

getch();

}

void modify(int\*p)

{ int i;

for(i=0;i<10;i++,p++)

{

\*p=\*p\*3;

}

}

**2.**

#include<stdio.h>

#include<conio.h>

void modify(int [],int);

void main()

{

int a[10],i;

clrscr();

printf("ENTER THE ELEMENTS:");

for(i=0;i<10;i++)

{

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

}

modify(a,10);

printf("THE MODIFIED ELEMENTS ARE:");

for(i=0;i<10;i++)

{

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

}

getch();

}

void modify(int b[],int n)

{

int i;

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

{

b[i]=b[i]\*3;

}

}

**PRACTICAL-32**

**AIM:**

**1. (Pointers to Functions)**

**#include<stdio.h>**

**void display();**

**void main()**

**{**

**void (\*func\_ptr)();**

**clrscr();**

**func\_ptr=display;**

**printf("Address of functions display is %u\n",func\_ptr);**

**(\*func\_ptr)();**

**getch();**

**}**

**void display()**

**{**

**puts("By helping others, we help overselves!!");**

**}**

**2.**

char \*copy (char\*,char \*);

void main()

{

char \*str;

char source[] = "Kindness";

char target[10];

str=copy(target,source);

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

getch();

}

char \*copy(char \*t,char \*s)

{

char \* r;

r = t;

while(\*s!='\0')

{

\*t=\*s;

t++;

s++;

}

\*t='\0';

return(r);

}

**PRACTICAL-33**

**AIM:**

An automobile company has serial number engine parts starting from AA0 to FF9. The other characteristics of parts to be specified in structure are year of manufacturing, material and quantity manufactured.

1. Specify a structure to store information corresponding to part.
2. Write a program using pointer to retrieve information on parts with serial numbers between BB1 and CC6. **(Pointers and Structures)**

**PROGRAM CODE:**

#include<stdio.h>

struct engine

{

char srno[3];

int yom;

char material[10];

int qua;

}s[9],\*p;

int main()

{

int i;

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

{

printf("enter srno : ");

gets(s[i].srno);

fflush(stdin);

printf("\nenter yom : ");

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

fflush(stdin);

printf("\nenter material : ");

gets(s[i].material);

fflush(stdin);

printf("\nenter qua : ");

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

fflush(stdin);

}

for(p=s;p<=s+8;p++)

{

if(p->srno[0]=='B'||p->srno[0]=='C')

{

printf("srno is : ");

puts(p->srno);

printf("\nyear of manufecharing is : %d\n",p->yom);

printf("material is : ");

puts(p->material);

printf("\nquantiaty is : %d\n",p->qua);

}

}

return 0;

}

**PRACTICAL-34**

**AIM:**

Write a program that takes contents of a file and copy them into another file and print it on the screen. Use **feof ()** functions to detect the end of file and **ferror()** function to detect if there is an error in opening the file.

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

char ch;

FILE \*fptr1,\*fptr2;

fptr1=fopen("f1.txt","r");

if(fptr1=='\0')

{

printf("file could not open");

}

fptr2=fopen("f2.txt","w");

if(fptr2=='\0')

{

printf("file could not open");

}

while((ch=getc(fptr1)) != EOF)

{

putc(ch,fptr2);

printf("%c",ch);

}

fclose(fptr1);

fclose(fptr2);

}

**PRACTICAL-35**

**AIM:**

Write a program to create a file named ALPHABETS which consists of all 26 letters ABC…XYZ and prints the contents of the file in reverse order ZYX….CBA on the screen. Use the function **ftell(), fseek() and rewind().**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int i;

char ch;

FILE \*fptr;

fptr=fopen("ALPHABET.txt","r");

if(fptr=='\0')

{

printf("ALPHABET.txt could not open.");

}

fseek(fptr,1,2);

for(i=0;i<27;i++)

{

ch=getc(fptr);

printf("%c",ch);

fseek(fptr,-2,1);

}

}

**PRACTICAL-H7**

**AIM:**

Write a program to open a file name INVENTORY and store in it the following data. Use **fprintf()** and **fscanf()** functions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Name** | **Number** | **Price** | **Quantity** |
| AAA1 | 111 | 17.5 | 100 |
| BBB2 | 125 | 35 | 50 |
| CCC3 | 150 | 50 | 200 |

**PROGRAM CODE:**

#include<stdio.h>

struct invent\_record

{

char name[10];

int number;

float price;

int qty;

};

int main()

{

struct invent\_record item;

char filename[10];

int response;

FILE \*fp;

long n;

printf("\t\t\t\tPRACTICAL H7\n\n");

void append(struct invent\_record \*x, FILE \*y);

printf("Enter filename: ");

scanf("%s",filename);

fp=fopen(filename,"a+");

do

{

append(&item,fp);

printf("\nItem %s appended.\n",item.name);

printf("\nDo you want to add another item ? (1 for YES / 0 for NO): ");

scanf("%d",&response);

}

while(response==1);

n=ftell(fp);

fclose(fp);

fp=fopen(filename,"r");

while(ftell(fp)<n)

{

fscanf(fp,"%s %d %f %d",item.name,&item.number,&item.price,&item.qty);

fprintf(stdout,"%-8s %7d %8.2f %8d\n",item.name,item.number,item.price,item.qty);

}

fclose(fp);

return 0;

}

void append(struct invent\_record \*product, FILE \*ptr)

{

printf("Item name: ");

scanf("%s",product->name);

printf("Item number: ");

scanf("%d",&product->number);

printf("Item price: ");

scanf("%f",&product->price);

printf("Quantity: ");

scanf("%d",&product->qty);

fprintf(ptr,"%s %d %.2f %d",product->name,product->number,product-

>price,product->price,product->qty);

}

**OUTPUT:**

**PRACTICAL-36**

**AIM:**

Two files Data1 and Data2 contains sorted list of integers. Write a program to produce file Data3 which holds a single sorted, merge list of these two list. Use **command line argument** to specify the file name.

**PROGRAM CODE:**

#include<stdio.h>

int main(int argc, char\*argv[])

{

FILE \*f,\*f1,\*f2;

int i,j,a[20],b[20];

f1=fopen("Data1.txt","r");

f2=fopen("Data2.txt","r");

f=fopen(argv[1],"w");

for(i=0;i<10;i++)

fscanf(f1,"%d",&a[i]);

for(i=0;i<10;i++)

fscanf(f2,"%d",&b[i]);

i=0;j=0;

while(i!=10 && j!=10)

{

if(a[i]<b[j])

{

fprintf(f," %d",a[i]);

i++;

}

else

{

fprintf(f," %d",b[j]);

j++;

}

}

while(i<10)

{

fprintf(f," %d",a[i]);

i++;

}

while(j<10)

{

fprintf(f," %d",b[i]);

j++;

}

return 0;

}

**PRACTICAL-37**

**AIM:**

Write a program to enter N numbers into array and sort the second half of the array using **function sort().** Enter the size of the array through keyboard. (**Dynamic Array**). Use **malloc ()** to allocate memory and use **free()** to free the memory after the use.

For example if input is

5 13 24 67 45 34

Output should be

5 13 24 **34 45 67**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

int n,\*a,\*y,i;

printf("enter size of element : ");

scanf("%d",&n);

a=(int \*)malloc(n\*sizeof(int));

printf("enter elements : \n");

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

{

scanf("%d",a+i);

}

sort(a,n);

return 0;

}

sort(int \*a,int n)

{

int i,j,temp;

for(i=n/2;i<n;i++)

{

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

{

if(\*(a+i)>\*(a+j))

{

temp=\*(a+i);

\*(a+i)=\*(a+j);

\*(a+j)=temp;

}

}

}

printf("sorted elements : \n");

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

{

printf("%d\n",\*(a+i));

}

}

**PRACTICAL-38**

**AIM:**

Write a program using to store a character string in a block of memory space created by **calloc ()** and then modify the same to store a larger string using **realloc ()** function. **(Dynamic Array).**

**PROGRAM CODE:**

#include<stdio.h>

int main()

{

char \*b;

b=(char \*)malloc(10);

if(b==NULL)

{

printf("allocation is failed");

exit(1);

}

strcpy(b,"charusat");

printf("%s",b);

b=realloc(b,30);

if(b==NULL)

{

printf("allocation is failed");

exit(1);

}

strcpy(b,"\ncharusat is best........");

printf("%s",b);

free(b);

}