**1.PRINT STRING X PATTERN:**

**2.TO FIND THE MATCH BETWEEN STRINGS AND CHECK IF THE STRING IS A SUBSTRING OF ANOTHER**

**INPUT1:TEST123**

**INPUT2:123**

**OUTPUT:4(INDEX OF STARTING SUBSTRING)**

**3.TO PRINT ODD ELEMENTS IN A ARRAY FIRST FOLLOWED BY EVEN ELEMENTS**

**4.APTITUDE POLICE:RUNS 1KM IN 1ST DAY 2KM IN 2ND DAY AND SO ON**

**THEIF:RUNS 30 KM EVERY DAY**

**FIND HOW MANY DAYS POLICE REQUIRED TO CATCH A THEIF**

**ANSWER:N=2M-1**

**5. TO FIND INDEX OF HIGHEST NEGATIVE ELEMENT IN ARRAY**

**INPUT: 1 -2 3 4 -5**

**OUTPUT:1**

#include<stdio.h>

#include<conio.h>

void main()

{

int \*a,min,i,j,n,temp=0,b,\*c;

clrscr();

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

scanf("%d",&n);

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

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

printf("enter the elements:");

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

{

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

}

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

{

c[i]=a[i];

}

printf(" original c array:");

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

{

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

}

min=a[0];

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

{

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

{

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

{

b=a[i];

a[i]=a[j];

a[j]=b;

}

}

}

printf("sorted c array is:");

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

{

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

}

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

{

if(min>a[i])

{

min=a[i];

j=i;

}

}

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

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

{

if(a[i]<0 && a[i]>min && j!=i)

{

temp=a[i];

}

if(min==a[i])

{

temp=a[i];

}

}

printf("\ntemp:%d",temp);

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

{

if(temp==c[i])

{

printf("temp at position at :%d",i);

}

}

getch();

}

**6.MERGE TWO ARRAY WITHOUT DUPLICATE ELEMENTS**

**7.TO PRINT COLUMN NAME OF EXCEL SHEET**

**EG: 28=AB**

**8.WEIGHT AND ITEM PGM AND SORT ITEMS BASED ON WEIGHT IN INCREASING ORDER**

**9.TO CHECK VALID PARANTHESIS**

#include<stdio.h>

**int** main(){

**char** a[24],stck[24];

**int** n,i,top=0,cnt=0;

scanf("%s",a);

n = strlen(a);

**if**(n%2==0){

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

**switch**(a[i]){

**case** '(':

**case** '{':

**case** '[':

**case** '<':stack[top++]=a[i];**break**;

**case** ')':**if**(stack[--top]=='('){ cnt++; }**break**;

**case** '}':**if**(stack[--top]=='{') cnt++; **break**;

**case** ']':**if**(stack[--top]=='[') cnt++; **break**;

**case** '>':**if**(stack[--top]=='<') cnt++; **break**;

**default**:**break**;

}

}

//printf("%s %d %d",stack,cnt,n);

**if**(n/2==cnt)

printf("valid");

**else**

printf("invalid");

}**else**

printf("invalid");

**return** 200;

}