**/\*1) WAP to count number of bits to be changed after converting first number to second number using**

**bitwise operator.**

**EX: i/p: n1=8,n2=7**

**binary for n1=00001000**

**binary for n2=00000111**

**for the given example if we convert n1 to n2 then 4 bits we need to change.**

**o/p: 4**

**NOTE:Do the operation for 32 bits.\*/**

/\*#include<stdio.h>

main()

{

int n1,n2,i,j,c;

printf("enter n1 and n2 ....\n");

scanf("%d%d",&n1,&n2);

for(i=31;i>=0;i--)

{

if(n1>>i&1!=n2>>i&1)

c++;

}

printf("number of bits to be changed after converting first number %d to second number %d =%d",n1,n2,c);

}\*/

**/\*2) WAP to print following pattern using function.**

**a**

**B c**

**D e F**

**g H i J**

**k L m N o**

**NOTE: Create generic code for pattern and use this prototype: void my\_fun(int).\*/**

#include<stdio.h>

void my\_fun(int);

main()

{

int r;

printf("enter your rows...\n");

scanf("%d",&r);

my\_fun(r);

}

void my\_fun(int r)

{

int i,j;

char ch='A';

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

{

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

{

if(j<=i)

{

printf("%c ",ch+j);

}

else

printf(" ");

}

ch=ch+i+1;

printf("\n");

}

}

**/\*3) WAP to search digit and it’s position from left side which is perfect number for given integer**

**number.**

**EX: i/p: n=4762591**

**o/p: from left side 3rd digit is perfect number which is 6.\*/**

/\*#include<stdio.h>

main()

{

int num,c=1,temp,sum=0;

printf("enter your number...\n");

scanf("%d",&num);

temp=num;

while(temp)

{

sum=sum\*10+temp%10;

temp=temp/10;

}

while(sum)

{

if(sum%10==6) **// single digit perfect number is only 6**

break;

c++;

sum=sum/10;

}

printf("from left side %d digit is perfect number which is 6",c);

}\*/

**/\*4) WAP to find common elements in three sorted array.**

**EX: a1= {1, 5, 10, 20, 40, 80}**

**a2= {6, 7, 20, 80, 100}**

**a3 = {3, 4, 15, 20, 30, 70, 80, 120}**

**o/p: 20, 80\*/**

/\*#include<stdio.h>

main()

{

int a1[6],a2[5],a3[8],ele1,ele2,ele3,i,j,k;

ele1=sizeof(a1)/sizeof(a1[0]);

ele2=sizeof(a2)/sizeof(a2[0]);

ele3=sizeof(a3)/sizeof(a3[0]);

printf("enter 1 array...\n");

for(i=0;i<ele1;i++)

scanf("%d",&a1[i]); // scanning 1st array

printf("enter 2 array...\n");

for(i=0;i<ele2;i++)

scanf("%d",&a2[i]); // scanning 2nd array

printf("enter 3 array...\n");

for(i=0;i<ele3;i++)

scanf("%d",&a3[i]); // scanning 3rd array

printf("a1:");

for(i=0;i<ele1;i++)

printf("%d ",a1[i]); // printing 1st array

printf("\n");

printf("a2:");

for(i=0;i<ele2;i++)

printf("%d ",a2[i]); // printing 2nd array

printf("\n");

printf("a3:");

for(i=0;i<ele3;i++)

printf("%d ",a3[i]);

printf("\n"); // printing 3rd array

for(i=0;i<ele1;i++) // lloop for a1 array

{

for(j=0;j<ele2;j++) // loop for a2 array

{

if(a1[i]==a2[j])

{

for(k=0;k<ele3;k++) // loop a3 array

{

if(a1[i]==a3[k])

printf("%d ",a1[i]);

}

}

}

}

}

\*/

**/\*5) WAP to reverse elements which are prime number from given array.**

**EX: i/p={10,13,20,17,24}**

**o/p={10,31,20,71,24}**

**Here in given example 13 and 17 are prime numbers so after reversing elements are 31 and 71\*/**

/\*#include<stdio.h>

int prime(int); // check prime number function prototype

main()

{

int i,max,j,a[5],ele,temp,sum;

ele=sizeof(a)/sizeof(a[0]);

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

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

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

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

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

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

{

temp=a[i];

j=prime(temp); // check wheather a prime or not

if(j==1)

{

sum=0;

while(temp) // reversing of number...

{

sum=sum\*10+temp%10;

temp=temp/10;

}

a[i]=sum; // assining reverse number...

}

}

printf("\nprime numbers so after reversing elements are\n");

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

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

}

int prime(int n)

{

int i;

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

{

if(n%i==0)

break;

}

if(n==i)

return 1;

}\*/

**/\*6) WAP to find pair in an array with largest product.**

**EX: a1={1, 0, 3, 2, 4}**

**o/p=3\*4=12\*/**

/\*#include<stdio.h>

main()

{

int i,max,j,a[5],ele,temp;

ele=sizeof(a)/sizeof(a[0]);

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

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

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

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

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

max=a[0]\*a[1];

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

{

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

{

temp=a[i]\*a[j];

if(temp>max)

max=temp;

}

}

printf("maximum mul=%d ",max);

}\*/

**/\*7) WAP to check if two given string is anagram of each other or not.**

**EX: a1=”Army”**

**a2=”Mary”**

**o/p=anagram string.**

**An anagram contains same length and same character, but in a different order.\*/**

/\*#include<stdio.h>

main()

{

char s[10],s1[10],i,j,c;

printf("enter your both string\n");

scanf("%s%s",s,s1);

for(i=0;s[i];i++)

{

for(j=0;s1[j];j++)

{

if((s[i]==s1[j]+32) || (s[i]==s1[j]-32))

{

s[i]=-1;

}

}

}

for(i=0;s[i];i++)

{

if(s[i]>0)

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

}

}\*/

**//8)WAP to check if a given String is Palindrome or not using recursive function.**

#include<stdio.h>

Void str\_check(char \*);

main()

{

char s[10],c[10],\*p,\*q,ch,i,j;

printf("enter your string..\n");

scanf("%s",s);

printf("enter string=%s",s);

for(i=0;s[i];i++)

c[i]=s[i];

c[i]=s[i];

p=q=s;

while(\*p)

p++;p--;

while(q<p)

{

ch=\*p;

\*p=\*q;

\*q=ch;

q++;

p--;

}

printf("\nrevesre string=%s",s);

for(i=0,j=0;s[i];j++,i++)

{

if(s[i]!=c[j])

break;

}

if(s[i]=='\0')

printf("\npalindrome string...");

else

printf("\nNOT palindrome string..");

}