1. Print the word with odd letters as

P M

R A

O R

G

O R

R A

P M

Ans:

#include<stdio.h>

#include<string.h>

int main()

{

char str[50];

char str2[50][50];

int lenstr;

int i,j;

char temp;

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

scanf("%s",str);

lenstr = strlen(str);

if(lenstr %2 == 0)

{

printf("The string length must be an odd length");

}

else

{

j = 0;

temp = 0;

for(i = 0;i == lenstr;i++)

{

str2[i][j] = str[i];

j = j + 1;

}

for(i = lenstr; i==0 ;i--)

{

j = lenstr;

str2[i][j] = str[temp];

temp = temp + 1;

j = j - 1;

}

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

{

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

{

printf("%c",str2[i][j]);

}

printf("\n");

}

}

return 0;

}

The output to the program must be for example: geeks

g g

e e

e

k k

s s

3. Save the string “WELCOMETOZOHOCORPORATION” in a two dimensional array and search for substring like “too” in the two dimensional string both from left to right and from top to bottom.

w e L C O

M E T O Z

O H O C O

R P O R A

T I O n

And print the start and ending index as

Start index : <1,2>

End index: <3, 2>

**Without 2d**

**#include<string.h>**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int c,k=0,i,j,rs,cs,r1,r2,c1,c2,t,flag=0,n,fail=1,sl,r;**

**char s[100],sub[100];**

**cin>>s;**

**cin>>sub;**

**cin>>c;**

**n=strlen (s);**

**sl=strlen (sub);**

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

**{**

**if(s[i]==sub[0])**

**{**

**rs=i;**

**for(i;k<sl;++i)**

**{**

**if(s[i]!=sub[k])**

**flag=1;**

**if(flag==1)**

**break;**

**++k;**

**}**

**if(flag==0)**

**{**

**fail=0;**

**r1=rs/c;**

**c1=rs%c;**

**--i;**

**r2=i/c;**

**c2=i%c;**

**cout<<r1<<","<<c1<<"\n"<<r2<<","<<c2;return 0;**

**}**

**else if(flag==1)**

**{**

**i=rs;flag=0,k=0;**

**for(i;k<sl;i+=c)**

**{**

**if(s[i]!=sub[k])**

**flag=1;**

**if(flag==1)**

**break;**

**++k;**

**}**

**if(flag==0)**

**{**

**fail=0;**

**r1=rs/c;**

**c1=rs%c;**

**i-=c;**

**r2=i/c;**

**c2=i%c;**

**cout<<r1<<","<<c1<<"\n"<<r2<<","<<c2;return 0;**

**}**

**else**

**{i=rs;k=0,flag=0;**

**}**

**}**

**}**

**}**

**if(fail==1)**

**cout<<"-1";**

**return 0;**

**}**

**With 2d**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**int c,k=0,i,j,rs,cs,t,flag=0,n,fail=0,r,sl;**

**char s[100],sub[100];**

**scanf("%s",&s);**

**scanf("%s",&sub);**

**scanf("%d",&c);**

**n=strlen (s);**

**sl=strlen (sub);**

**r=n/c;**

**if(n%c)**

**++r;**

**char a[r][c];**

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

**{**

**for(j=0;j<c;++j)**

**{**

**a[i][j]=s[k];**

**++k;**

**}**

**}**

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

**{**

**for(j=0;j<c;++j)**

**{**

**if (a[i][j]==sub[0])**

**{**

**t=1;rs=i;cs=j;++i;**

**for(i;((i<r)&&(t<sl));++i)**

**{**

**if(a[i][j]!=sub[t])**

**{**

**flag=1;**

**}**

**if(flag==1)**

**break;**

**++t;**

**}**

**if(flag==0)**

**{**

**printf("%d,%d\n%d,%d",rs,cs,--i,j);**

**return 0;**

**}**

**else if(flag==1)**

**{**

**flag=0;i=rs;j=cs;t=1;++j;**

**if(j<c)**

**{**

**for(j;((j<c)&&(t<sl));j++)**

**{**

**if(a[i][j]!=sub[t])**

**{**

**flag=1;**

**}**

**if(flag==1)**

**break;**

**++t;**

**}**

**}**

**else**

**flag=1;**

**}**

**if(flag==0)**

**{**

**printf("%d,%d\n%d,%d",rs,cs,i,--j);**

**return 0;**

**}**

**else**

**{**

**i=rs;j=cs;fail=1;//k=0,flag=0;**

**}**

**}**

**}**

**}**

**if(fail==1)**

**printf("-1");**

**return 0;**

**}**

1. Strring reverse
2. :main()

{

int i,n,j;

char a[100],t;

pf(enter a str)

sf(%s,a)

n=strlen(a)

for(i=0,j=n-1;i<n/2,j>=n/2;i++,j--)

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

pf(a);

}

1. Convert to binary:
2. #include<stdio.h>

#include<conio.h>

void main()

{

int a,b[100],c=0,i=0,n;

clrscr();

printf("\n enter the number");

scanf("%d",&a);

z :

if((a/2)!=0)

{

b[i]=a%2;

i++;

n=i;

a=a/2;

goto z;

}

else if((a/2)==0)

{

b[n]=a;

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

{

if(b[i]==1)

c=c+1;

}

}

printf("\n the bit set of %d is %d",a,c);

printf("\nBinary equivalent of %d is:",a);

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

{

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

}

getch();

}

1.Wite a code to implement M pattern using stars with N as input denoting the number of rows.

Eg. For N=3 or N=4 the respective output must be displayed.

|  |  |
| --- | --- |
| N=3 | N=4 |
| \* \*  \* \* \* \*  \* \* \* | \* \*  \* \* \* \*  \* \* \* \*  \* \* \* |

**BASIC PATTERN LOGIC PATTERN**

|  |
| --- |
| \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \* |

1.

**Program(Method 1):**

#include <stdio.h>

void main()

{

int i,j,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

{

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

printf("%s","\*");

printf("\n");

}

}

**Program(Method 2 – Best Method):**

#include <stdio.h>

void main()

{

char\* c = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

printf("%\*.\*s\n",i,i,c);

}

|  |
| --- |
| \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \* |

2.

**Program(Method 1):**

#include <stdio.h>

void main()

{

int i,j,k,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

{

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

printf("%s"," ");

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

printf("%s","\*");

printf("\n");

}

}

**Program(Method 2 – Better than method 1)**

#include <stdio.h>

void main()

{

int i,j,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

{

printf("%\*s",n-i,"");

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

printf("%s","\*");

printf("\n");

}

}

**Program(Method 3 – Best Method)**

#include <stdio.h>

void main()

{

char\* c = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

printf("%\*.\*s\n",n,i,c);

}

|  |
| --- |
| A  AB  ABC  ABCD  ABCDE |

3.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "ABCDEFGHIJKL";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

printf("%\*.\*s\n",n,i,c);

}

|  |
| --- |
| A  AB  ABC  ABCD  ABCDE |

4.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "ABCDEFGHIJKL";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

printf("%\*.\*s\n",i,i,c);

}

|  |
| --- |
| A  BC  DEF  GHIJ  KLMNO |

5.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "ABCDEFGHIJKLMNOPQRS";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

printf("%\*.\*s\n",i,i,c);

}

|  |
| --- |
| \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\* |

6.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);printf("\n");

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

printf("%\*.\*s\n",n+i,2\*i+1,c);

}

|  |
| --- |
| A  ABC  ABCDE  ABCDEFG  ABCDEFGHI |

7.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "ABCDEFGHIJKLM";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);printf("\n");

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

printf("%\*.\*s\n",n+i,2\*i+1,c);

}

|  |
| --- |
| \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*  \*\*\*  \* |

8.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

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

printf("%\*.\*s\n",n+i,2\*i+1,c);

for(i=n-2;i>=0;i--)

printf("%\*.\*s\n",n+i,2\*i+1,c);

}

|  |
| --- |
| Enter character : E  A  ABA  ABCBA  ABCDCBA  ABCDEDCBA |

9.

**PROGRAM:**

#include<stdio.h>

void main()

{

char ch,r,c;

printf("Enter character : ");

scanf("%c",&ch);printf("\n");

if(ch>='a' && ch<='z')

ch=ch-32;

for(r='A'; ch>=r; r++)

{

printf("%\*s",ch-r,"");

for(c='A'; r>=c; c++)

printf("%c",c);

for(c=r-1; c>='A'; c--)

printf("%c",c);

printf("\n");

}

}

|  |
| --- |
| \* \* \*  \* \* \* \*  \* \* \* \*  \* \* \* \*  \* \* \* \*  \* \* |

10.

**PROGRAM:**

#include<stdio.h>

void main()

{

int i,j,n,s,m,L;

printf("Enter the number of row in W");

scanf("%d",&n);printf("\n");

s=1;

m=2\*n-1;

L=2\*m-1;

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

{

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

if(((s+i) == j)||((m-i) == j)||((m+i) == j)||((L-i) == j))

printf("\*");

else

printf(" ");

printf("\n");

}

}

|  |
| --- |
| \*  \*A\*A\*  \*A\*A\*A\*A\*  \*A\*A\*A\*A\*A\*A\*  \*A\*A\*A\*A\*A\*A\*A\*A\* |

11.

**PROGRAM:**

#include <stdio.h>

void main()

{

char\* c = "\*A\*A\*A\*A\*A\*A\*A\*A\*A\*A\*A\*A\*A";

int i,n;

printf("Enter the Number of lines:");

scanf("%d",&n);

for(i=0;i<2\*n;i+=2)

{

printf("%\*.\*s\n",2\*n+i,2\*i+1,c);

}

}

|  |
| --- |
| Enter the char: Z  A A  AB BA  ABC CBA  ABCD DCBA  ABCDE EDCBA  ABCDEF FEDCBA  ABCDEFG GFEDCBA  ABCDEFGH HGFEDCBA  ABCDEFGHI IHGFEDCBA  ABCDEFGHIJ JIHGFEDCBA  ABCDEFGHIJK KJIHGFEDCBA  ABCDEFGHIJKL LKJIHGFEDCBA  ABCDEFGHIJKLM MLKJIHGFEDCBA  ABCDEFGHIJKLMN NMLKJIHGFEDCBA  ABCDEFGHIJKLMNO ONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOP PONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQ QPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQR RQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRS SRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRST TSRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRSTU UTSRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRSTUV VUTSRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRSTUVW WVUTSRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRSTUVWX XWVUTSRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRSTUVWXY YXWVUTSRQPONMLKJIHGFEDCBA  ABCDEFGHIJKLMNOPQRSTUVWXYZYXWVUTSRQPONMLKJIHGFEDCBA |

12.

#include <stdio.h>

void main()

{

int i,j;

char n;

char c ='A';

printf("Enter the Char:");

scanf("%d",&n);

for(i=65;i<=n;i++)

{

for(j=65;j<=i;j++)

{

printf("%c",c);

c++;

}

if(i!=n)

printf("%\*s",2\*(n-i)-1,"");

else

c--;

for(j=i;j>64;j--)

{

if(i==n)

if(j==65)

break;

printf("%c",--c);

}

printf("\n");

}

}

Pattern:

1

232

34543

4567654

567898765

C program:

#include<stdio.h>

main()

{

int n, c, d, num = 1, space;

scanf("%d",&n);

space = n - 1;

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

{

num = d;

for ( c = 1 ; c <= space ; c++ )

printf(" ");

space--;

for ( c = 1 ; c <= d ; c++ )

{

printf("%d", num);

num++;

}

num--;

num--;

for ( c = 1 ; c < d ; c++)

{

printf("%d", num);

num--;

}

printf("**\n**");

}

return 0;

}

Write a program to give the following output for the given input

Eg 1: Input: a1b10

Output: abbbbbbbbbb

Eg: 2: Input: b3c6d15

Output: bbbccccccddddddddddddddd

The number varies from 1 to 99.

**MY SOLUTION ::**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

**int** **main**() {

**char** s[10];

**int** i = 0;

**int** count,j;

**int** inc;

scanf("%s",s);

**while** (1==sscanf(s+i, "%\*[^0-9]%d%n", &count, &inc))

{

**for**(j=0;j<count;j++)

printf("%c",\*(s+i));

i += inc;

count = 1;

}

**return** 0;

}

[Zoho Interview | Set 2 Question 3](http://markandayannotes.blogspot.com/2015/08/zoho-interview-set-2-question-3.html)

Form a number system with only 3 and 4. Find the nth number of the number system.  
  
Eg.) The numbers are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343, 3344, 3433, 3434, 3443, 3444 ….  
  
  
**My Solution::**

#include<stdio.h>

#include<conio.h>

**void** bin(**int** n){

**if**(n>1)

bin(n/2);

printf("%d",(n%2)?3:4);

}

**int** **main**(){

**int** num,i;

clrscr();

scanf("%d",&num);

i = 1;

**do**{

num -= 1<<i;

i++;

}**while**(num);

--i;

--i;

num += 1<<i;

bin(num);

getch();

**return** 0;

}

1. Print the pattern  
   N=4  
   4444444  
   4333334  
   4322234  
   4321234  
   4322234  
   4333334  
   4444444

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int n,m[100][100],a,b;

cout<<"enter the number:";

cin>>n;

int N=(2\*n)+1;

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

{

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

{

int a=(abs (n-i));

int b=(abs (n-j));

if (a>=b)

m[i][j]=a;

else

m[i][j]=b;

cout<<m[i][j];

}

cout<<"\n";

}

}

**Convert numbers to roman numerals in c**

#include<stdio.h>

void predigits(char c1,char c2);

void postdigits(char c,int n);

char roman\_Number[1000];

int i=0;

int main(){

    int j;

    long int number;

    printf("Enter any natural number: ");

    scanf("%d",&number);

    if(number <= 0){

         printf("Invalid number");

         return 0;

    }

    while(number != 0){

         if(number >= 1000){

             postdigits('M',number/1000);

             number = number - (number/1000) \* 1000;

         }

         else if(number >=500){

             if(number < (500 + 4 \* 100)){

                 postdigits('D',number/500);

                 number = number - (number/500) \* 500;

             }

             else{

                 predigits('C','M');

                 number = number - (1000-100);

             }

         }

         else if(number >=100){

             if(number < (100 + 3 \* 100)){

                 postdigits('C',number/100);

                 number = number - (number/100) \* 100;

             }

             else{

                 predigits('L','D');

                 number = number - (500-100);

             }

         }

         else if(number >=50){

             if(number < (50 + 4 \* 10)){

                 postdigits('L',number/50);

                 number = number - (number/50) \* 50;

             }

             else{

                 predigits('X','C');

                 number = number - (100-10);

             }

         }

         else if(number >=10){

             if(number < (10 + 3 \* 10)){

                 postdigits('X',number/10);

                 number = number - (number/10) \* 10;

             }

             else{

                 predigits('X','L');

                 number = number - (50-10);

             }

         }

         else if(number >=5){

             if(number < (5 + 4 \* 1)){

                 postdigits('V',number/5);

                 number = number - (number/5) \* 5;

             }

             else{

                 predigits('I','X');

                 number = number - (10-1);

             }

         }

         else if(number >=1){

             if(number < 4){

                 postdigits('I',number/1);

                 number = number - (number/1) \* 1;

             }

             else{

                 predigits('I','V');

                 number = number - (5-1);

             }

         }

    }

    printf("Roman number will be: ");

    for(j=0;j<i;j++)

         printf("%c",roman\_Number[j]);

    return 0;

}

void predigits(char c1,char c2){

    roman\_Number[i++] = c1;

    roman\_Number[i++] = c2;

}

void postdigits(char c,int n){

    int j;

    for(j=0;j<n;j++)

         roman\_Number[i++] = c;

}

**Sample output:**

Enter any natural number: 87

Roman number will be: LXXXVII

**C program for palindrome without using string functions**

#include <stdio.h>

#include <string.h>

int main()

{

char text[100];

int begin, middle, end, length = 0;

gets(text);

while (text[length] != '**\0**')

length++;

end = length - 1;

middle = length/2;

for (begin = 0; begin < middle; begin++)

{

if (text[begin] != text[end])

{

printf("Not a palindrome.**\n**");

**break**;

}

end--;

}

if (begin == middle)

printf("Palindrome.**\n**");

return 0;

}