**PROGRAMS USING RECURSION**

Q1)

#include<stdio.h>

void print(i)

{

if (i>50)

return;

else

{

printf("%d ",i);

print(i+1);

}

}

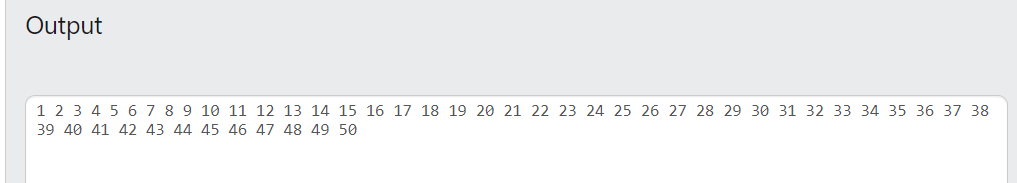
int main()

{

print(1);

return 0;

}



Q2)

#include<stdio.h>

void sum(int n)

{

static s=0;

if (n<0)

printf("%d",s);

else

{

s+=n;

sum(n-1);

}

}

int main()

{

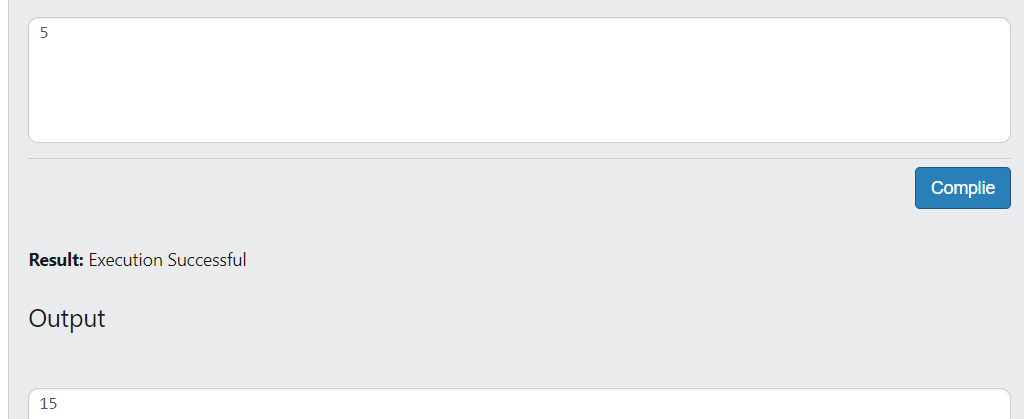
int n;

scanf("%d",&n);

sum(n);

return 0;

}



Q3)

#include<stdio.h>

int fibonacci(int n)

{

if (n==1)

{

return 1;

}

else if(n==2)

{

return 1;

}

else

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

}

int main()

{

int n,i;

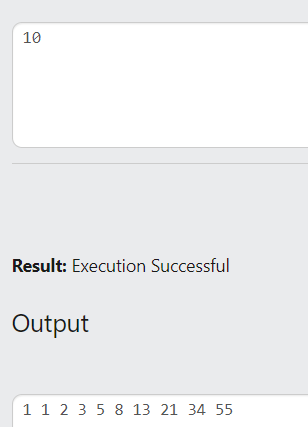
scanf("%d",&n);

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

printf("%d ",fibonacci(i));

return 0;

}



Q4)

#include<stdio.h>

void print(int a[],int n)

{

if (n==0)

{

return;

}

else

{

print(a,n-1);

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

}

}

int main()

{

int n,i;

scanf("%d",&n);

int a[n];

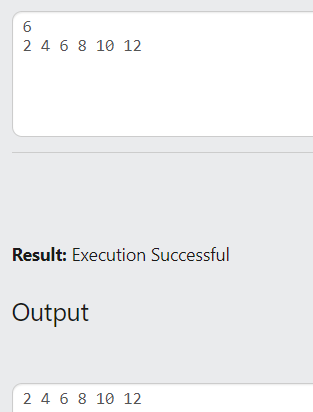
for (i=0;i<n;i++)

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

print(a,n);

return 0;

}



Q5)

#include<stdio.h>

void print(int a[],int n)

{

if (n==0)

{

return;

}

else

{

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

print(a,n-1);

}

}

int main()

{

int n,i;

scanf("%d",&n);

int a[n];

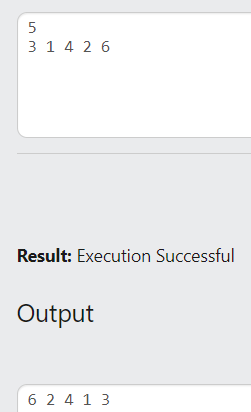
for (i=0;i<n;i++)

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

print(a,n);

return 0;

}



Q6)

#include<stdio.h>

int digit(int n)

{

static int c=0;

if (n==0)

{

return c;

}

else

{

c++;

digit(n/10);

}

}

int main()

{

int n,i,count;

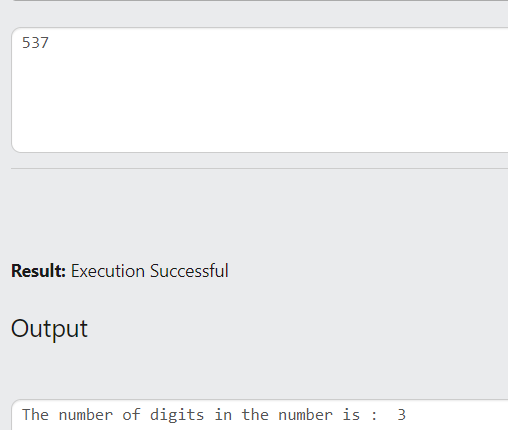
scanf("%d",&n);

count=digit(n);

printf("The number of digits in the number is : %d",count);

return 0;

}



Q7)

#include<stdio.h>

int digit(int n)

{

static int s=0;

if (n==0)

{

return s;

}

else

{

s+=n%10;

digit(n/10);

}

}

int main()

{

int n,sum;

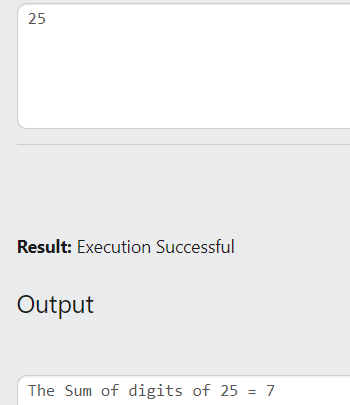
scanf("%d",&n);

sum=digit(n);

printf("The Sum of digits of %d = %d",n,sum);

return 0;

}



Q8)

#include<stdio.h>

int m=0;

void max(int a[],int n)

{

if (n==0)

printf("Largest element of an array is:%d",m);

else

{

if (a[n-1]>m)

m=a[n-1];

max(a,n-1);

}

}

int main()

{

int n,i;

scanf("%d",&n);

int a[n];

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

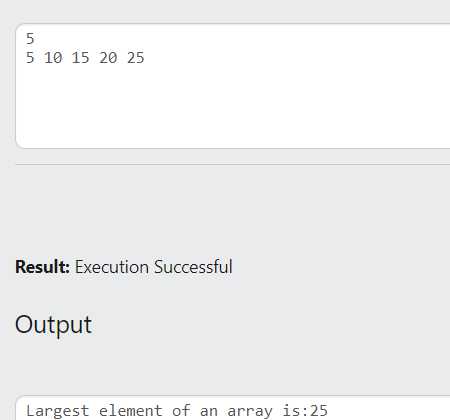
{

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

}

max(a,n);

}



Q9)

#include<stdio.h>

#include<string.h>

char s1[50];

int i=0;

void reverse(char s[],int len)

{

if (len==0)

puts(s1);

else

{

s1[i]=s[len-1];

i++;

reverse(s,len-1);

}

}

int main()

{

int len;

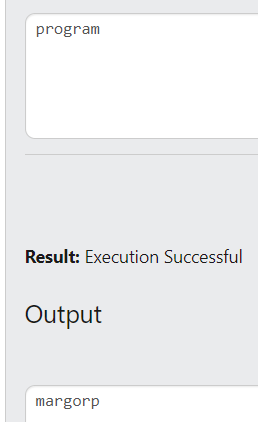
char s[50];

gets(s);

len=strlen(s);

reverse(s,len);

}



Q10)

#include<stdio.h>

int fact(n);

int main()

{

int n,factorial;

scanf("%d",&n);

factorial=fact(n);

printf("The Factorial of %d is : %d",n,factorial);

}

int fact(n)

{

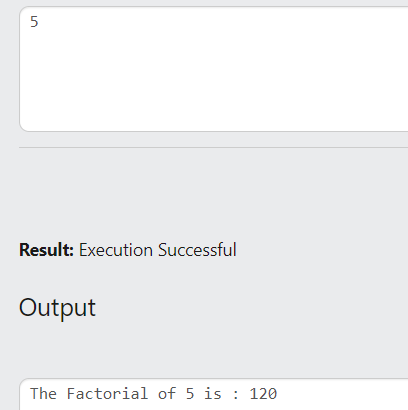
if (n==0)

return 1;

else

return n\*fact(n-1);

}



Q11)

#include<stdio.h>

void binary(n);

int main()

{

int n;

scanf("%d",&n);

binary(n);

}

void binary(n)

{

if (n==0)

return;

else

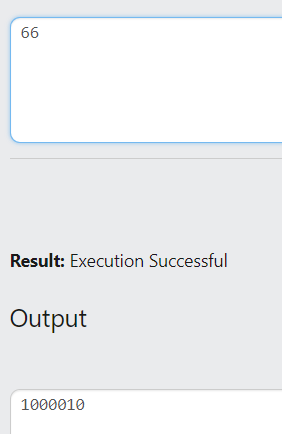
{

binary(n/2);

printf("%d",n%2);

}

}



Q12)

#include<stdio.h>

void odd(int s,int e);

void even(int s,int e);

int main()

{

int n1,n2;

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

if (n1%2==0)

{

printf("All even numbers from %d to %d are : ",n1,n2);

even(n1,n2);

printf("\nAll odd numbers from %d to %d are : ",n1,n2);

odd(n1+1,n2);

}

else

{

printf("All even numbers from %d to %d are : ",n1,n2);

even(n1+1,n2);

printf("\nAll odd numbers from %d to %d are : ",n1,n2);

odd(n1,n2);

}

}

void odd(int s,int e)

{

if (s>e)

return;

else

{

printf("%d ",s);

odd(s+2,e);

}

}

void even(int s,int e)

{

if (s>e)

return;

else

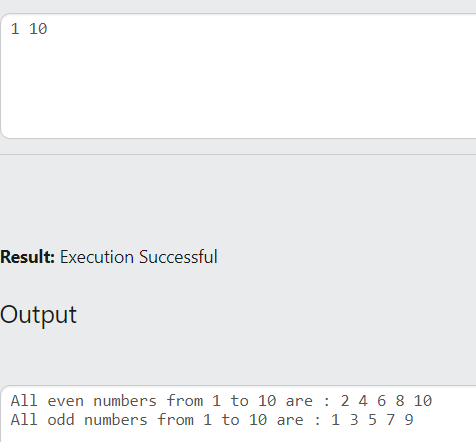
{

printf("%d ",s);

even(s+2,e);

}

}



Q13)

#include<stdio.h>

#include<string.h>

char s1[50];

int i=0;

void reverse(char s[],int len)

{

if (len==0)

{

//puts(s1);

return;

}

else

{

s1[i]=s[len-1];

i++;

reverse(s,len-1);

}

}

int main()

{

int len;

char s[50];

gets(s);

len=strlen(s);

reverse(s,len);

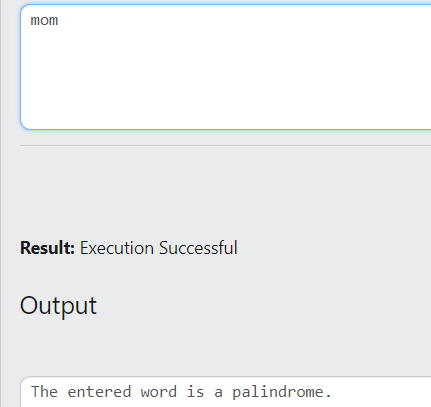
if (strcmp(s1,s)==0)

printf("The entered word is a palindrome. ");

else

printf("The entered word is not a palindrome. ");

}



Q14)

#include<stdio.h>

int power(int a,int b)

{

if (b==0)

return 1;

else

return a\*power(a,b-1);

}

int main()

{

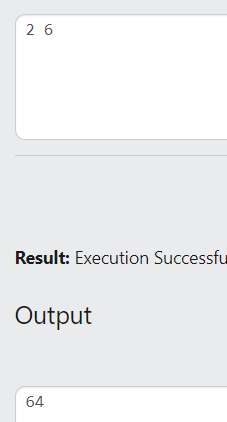
int n1,n2,v;

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

v=power(n1,n2);

printf("%d",v);

}



Q15)

#include<stdio.h>

#include<string.h>

char s2[50];

void copy(char s1[],int n);

int main()

{

int len;

char s1[50];

gets(s1);

len=strlen(s1);

copy(s1,len);

printf("The first string is : %s\n",s1);

printf("The copied string is : %s",s2);

}

void copy(char s1[],int n)

{

if (n==0)

{

//puts(s2);

return;

}

else

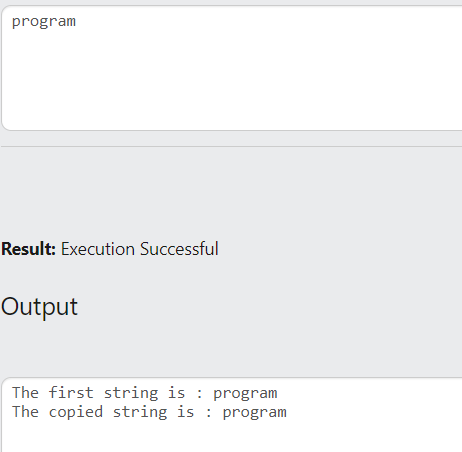
{

s2[n-1]=s1[n-1];

copy(s1,n-1);

}

}



Q16)

#include<stdio.h>

int pos=-1;

int linearsearch(int a[],int n,int e)

{

if (n<0)

return pos;

else if(a[n-1]==e)

{

pos=n-1;

return pos;

}

else

{

linearsearch(a,n-1,e);

}

}

int main()

{

int n,i,e,v;

scanf("%d",&n);

int a[n];

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

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

scanf("%d",&e);

v=linearsearch(a,n,e);

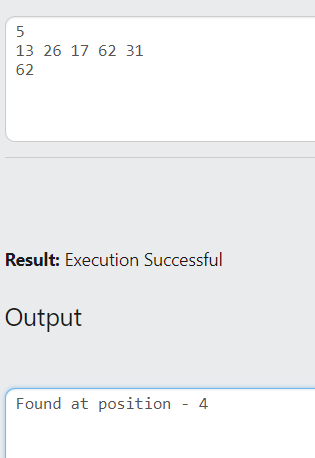
if (v==-1)

printf("Element Not Found");

else

printf("Found at position - %d",v+1);

}



Q17)

#include<stdio.h>

int binarysearch(int a[],int f,int l,int e)

{

int mid;

mid=(f+l)/2;

if (f>l)

return -1;

else

{

if(a[mid]==e)

return mid;

else if(e<a[mid])

return binarysearch(a,f,mid-1,e);

else

return binarysearch(a,mid+1,l,e);

}

}

int main()

{

int n,i,e,v;

scanf("%d",&n);

int a[n];

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

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

scanf("%d",&e);

v=binarysearch(a,0,n,e);

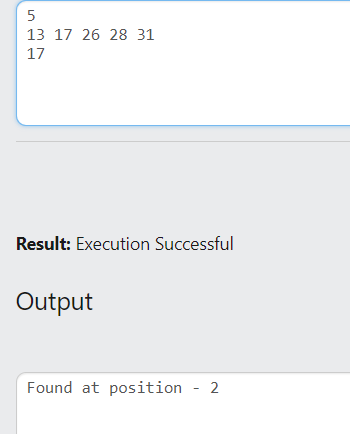
if (v==-1)

printf("Element Not Found");

else

printf("Found at position - %d",v+1);

}



Q18) Check whether given array is ascending or descending using recursion.

A)

#include<stdio.h>

int flag=0;

int ascending(int a[],int n)

{

if (n==1)

return flag;

else

{

if (a[n-1]<a[n-2])

flag=1;

ascending(a,n-1);

}

}

int main()

{

int n,i,v;

scanf("%d",&n);

int a[n];

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

{

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

}

v=ascending(a,n);

if (v==1)

printf("NO");

else

printf("YES");

}

Q19) Print array elements in reverse order.

A) #include<stdio.h>

void print(int a[],int n)

{

if (n==0)

return;

else

{

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

print(a,n-1);

}

}

int main()

{

int n,i;

scanf("%d",&n);

int a[n];

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

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

print(a,n);

}