Recursion

It is possible for the function to call themselves.

rec(int);

Void main()

{

int a ,fact;

printf(“\n enter any no”);

scanf(“%d”,&a);

fact=rel(a);

printf(“Factorial value=%d”,fact);

}

rec(int x)

{

If(x==1)

{

return(1);

else

f=x\*rec(x-1);

return(f);

}

rec(x)

{

=x\*rec(x-1); 2

3\* 3-1

=x\*rec(x-1);=1

2\* 2-1

=x\*rec(x-1);

|  |  |  |
| --- | --- | --- |
| Int fact(int x)  {int f;  if(x==1||x==0)  return 1;  else  {  f=x\*fact(x-1);  return f;}  }  To main() | Int fact(int x)  {int f;  if(x==1|| x==0)  return 1;  else  { f=x\*fact(x-1);  return f;}  } | Int fact(int x)  {int f;  if(x==1|| x==0)  return 1;  else  { f=x\*fact(x-1);  return f;  }  } |

• A recursive function is a function that invokes itself

directly or indirectly. For a recursive function to terminate,

there must be one or more base cases.

• Recursion is an alternative form of program control. It

is essentially repetition without a loop control. It

can be used to specify simple, clear solutions for

inherently recursive problems that would otherwise be

difficult to solve.

• Sometimes the original function needs to be modified to

receive additional parameters in order to be invoked

recursively. A recursive helper function can be

declared for this purpose.

• Recursion bears substantial overhead. Each time the

program calls a function, the system must assign space

for all of the function’s local variables and

parameters. This can consume considerable memory

8.3

8.4

8.5

8.6