1. What is a direct recursion?

A function that calls itself repetitively.

Direct recursion occurs when a function calls itself. This results in a one-step recursive call: the function makes a recursive call inside its own function body.

1. For the remaining questions, use the following recursive function:

| int Smiley (int n){  if (n == 0)  return 3;  else  return n + Smiley(n-1);  } |
| --- |

* + - What final value is returned from the function call Smiley(5)?

n=5

false

5 + Smiley(4)

n=4

false

4 + Smiley(3)

n=3

false

3 + Smiley(2)

n=2

false

2 + Smiley(1)

n=1

false

1 + Smiley(0)

n=0

true

3

#include <stdio.h>

int Smiley (int n);

int main()

{

int num;

printf(“Enter a number:”);

scanf(“%d”,&num);

printf(“Total: %d”, Smiley (num));

return 0;

}

int Smiley (int n)

{

if (n == 0)

return 3;

else

return n + Smiley(n-1);

}

* + - What final value is returned from the function call Smiley(0)?

3

* + - What final value is returned from the function call Smiley(-1)?

infinite loop toward negative infinity

* + - Suppose + is changed to \* in the recursive step. What is the answer for Smiley(5)?

360

1. Is the following function circular? Explain.

int Thor(int pq)

{ if (pq==1)

return 0;

else if (pq % 2 != 0)

return Thor(pq/2);

else return 1 + Thor(3\*pq + 1);}

pq=1

0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pq=2

false

false

1 + Thor(3\*2 + 1) =7

pq=7

false

true

Thor(7/2) =3

pq=3

false

true

Thor(3/2) =1

pq=1

0

output=1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pq=3

false

true

Thor(3/2) =1

pq=1

0

output=0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

pq=13

false

true

Thor(13/2) =6

pq=6

false

false

1 + Thor(3\*6 + 1) =19

pq=19

false

true

Thor(19/2) =9

pq=9

false

true

Thor(9/2) =4

pq=4

false

false

1 + Thor(3\*4 + 1) =13

circular function. repeat infinitely for certain numbers.

4. Consider the following program, (*If there is an error in the code and you need to correct it accurately*)

#include <stdio.h>

int power(int n1, int n2);

int main() {

int a, result, base;

printf("Enter base number: ");

scanf("%d", &base);

printf("Enter power number(positive integer): ");

scanf("%d", &a);

result = power(base, a);

printf("%d^%d = %d", base, a, result);

return 0;

}

int power(int base, int a, int result) {

if (a != 0)

return (power \* pow(base, a + 1));

else

return 1;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

int power(int n1, int n2);

int main() {

int a, result, base;

printf("Enter base number: ");

scanf("%d", &base);

printf("Enter power number(positive integer): ");

scanf("%d", &a);

result = power(base, a);

printf("%d^%d = %d", base, a, result);

return 0;

}

int power(int base, int a, int result) {

if (a != 0)

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

else

return 1;

}

5. Write a program in C to print first 50 natural numbers using recursion.

The natural numbers are : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50.

Amirul

copy paste

Ikmal:

#include <stdio.h>

int num(int z);

int main()

{

int z;

z=1;

num(z);

return 0;

}

int num(int z)

{

if(z<50)

{

printf("%d ", z);

num(z+1);

}

else

{

printf("%d",z);

}

}

Khairi:

#include <stdio.h>

int num(int n);

int main()

{

int n;

n=1;

num(n);

return 0;

}

int num(int n)

{

if(n<50)

{

printf("%d. ", n);

num(n+1);

}

else

{

printf("%d",n);

}

}

Haiqal :

#include <stdio.h>

int num(int n);

int main()

{

int n=1;

printf ("The natural numbers are : ");

num(n);

return 0;

}

int num(int n)

{

if(n<50)

{

printf("%d ", n);

num(n+1);

}

else

{

printf("%d",n);

}

}

Ain

#include <stdio.h>

int number(int n);

int main()

{

int n;

n=1;

number(n);

return 0;

}

int number(int n)

{

if(n<50)

{

printf("%d ", n);

number(n+1);

}

else

{

printf("%d",n);

}

}

Anis:

#include <stdio.h>

int number(int n);

int main()

{

int n;

n=1;

number(n);

return 0;

}

int number(int n)

{

if(n<50)

{

printf("%d ", n);

printf(“\t”);

number(n+1);

}

else

{

printf("%d",n);

}

}

elvis:

#include <stdio.h>

int num(int z);

int main()

{

int z;

z=1;

num(z);

scanf("%d",&z);

printf("%d",num(z));

return 0;

}

int num(int z)

{

if(z<50)

{

printf("%d ", z);

num(z+1);

}

else

{

printf("%d",z);

}

}

Khadijah:

#include<stdio.h>

int fun(int num);

int main()

{

int num=1;

fun(num);

return 0;

}

int fun(int num)

{

if (num==51)

return;

else

printf("%d\t", num);

return fun(num+1);

}

qIkhzam

#include<stdio.h>

void num(int ikmal , int hakim);

int main()

{

int n,p=1;

printf("Saya Ikmal, Masukkan Nombor :");

scanf("%d",&n);

num(n,p);

return 0;

}

void num(int n , int p)

{

if(n>p)

{

printf("%d\t",p);

num(n,p+1);

}

else

{

printf("%d\t",n);

}

}