

DSA LAB-3

J.Akash Siva Teja

22BCE1778

Submitted To:

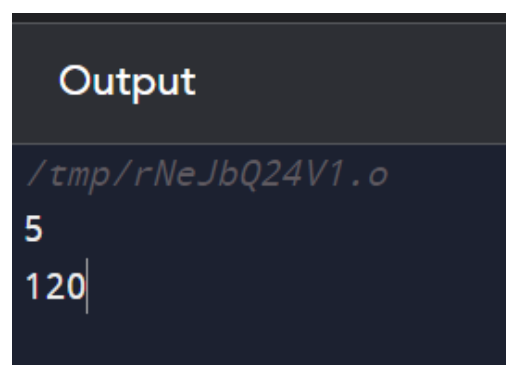
Dr.Ramesh Ragala

FACTORIAL:

CODE:

```
#include <stdio.h>
//factorial using recursive functions
int Rfact(int n) {
    if(n==1) {
        return 1;
    }
    else {
        return(n*Rfact(n-1));
    }
}
int main()
{
    int n,ans;
    scanf("%d",&n);
    ans = Rfact(n);
    printf("%d",ans);
    return 0;
}
```

OUTPUT:

A screenshot of a terminal window with a dark background. The title bar at the top says "Output". The terminal shows the command prompt `/tmp/rNeJbQ24V1.o` followed by the input `5`. The program then outputs `120`, and a cursor is visible at the end of the line.

```
Output
/tmp/rNeJbQ24V1.o
5
120|
```

ALGORITHM:

Algorithm Factorial(n: integer)

```
{  
    if n = 0 {  
        return 1  
    }  
    else{  
        return n * Factorial(n - 1)  
    }  
}
```

FIBONACCI SERIES:

CODE:

```
#include <stdio.h>

int Rfib(int n) {
    if(n==0) {
        return 0;
    }
    else if(n==1) {
        return 1;
    }
    else if(n==2) {
        return 1;
    }
    else {
        return(Rfib(n-1)+Rfib(n-2));
    }
}

int main()
{
    int n;
    scanf("%d",&n);
    int res = Rfib(n);
    for(int i=0;i<n;i++) {
        printf("%d ", Rfib(i));
    }
    return 0;
}
```

OUTPUT:

```
Output
/tmp/rNeJbQ24V1.o
10
0 1 1 2 3 5 8 13 21 34 |
```

ALGORITHM:

Algorithm Fibonacci(n)

```
{
    if n = 0 {
        return 0
    }
    else if n = 1 {
        return 1
    }
    else if n = 2 {
        return 1
    }
    else {
        return Fibonacci(n - 1) + Fibonacci(n - 2)
    }
}
```

GCD:

CODE:

```
#include <stdio.h>

int GCD(int x,int y) {
    if(x==0) {
        return y;
    }
    else if(y==0) {
        return x;
    }
    else {
        if(x>y) {
            return(x%y);
        }
        else {
            return(y%x);
        }
    }
}

int main()
{
    int x,y;
    scanf("%d %d",&x,&y);
    int res = GCD(x,y);
    printf("%d",res);
    return 0;
}
```

OUTPUT:

```
Output
/tmp/rNeJbQ24V1.o
5 3
2
```

ALGORITHM:

Algorithm GCD(x,y)

```
{
    if x = 0 {
        return y
    }
    else if y = 0 {
        return x
    }
    else {
        if x > y {
            return GCD(x % y, y)
        }
        else {
            return GCD(x, y % x)
        }
    }
}
```

RABBIT POPULATION:

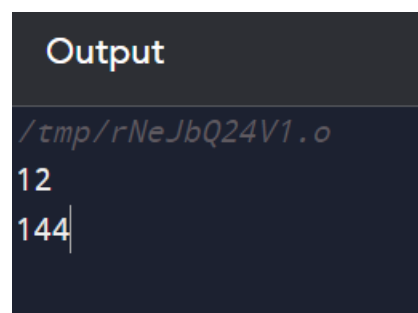
CODE:

```
#include <stdio.h>

int R_Rabbit(int n) {
    if(n==1) {
        return 1;
    }
    else if(n==2) {
        return 1;
    }
    else {
        return R_Rabbit(n-1)+R_Rabbit(n-2);
    }
}

int main() {
    int n;
    scanf("%d",&n);
    int r = R_Rabbit(n);
    printf("%d",r);
    return 0;
}
```

OUTPUT:



```
Output
/tmp/rNeJbQ24V1.o
12
144
```


ALGORITHM:

Algorithm R_Rabbit(n)

```
{  
    if n = 1 {  
        return 1  
    }  
    else if n = 2 {  
        return 1  
    }  
    else {  
        return R_Rabbit(n - 1) + R_Rabbit(n - 2)  
    }  
}
```

TOWERS OF HANOI:

CODE:

```
#include<stdio.h>

void towersofhanoi(int n,char a,char b,char c)
{
    if(n>=1)
    {
        towersofhanoi(n-1,a,c,b);
        printf("move disk no %d to tower %c\n",n,b);
        towersofhanoi(n-1,c,b,a);
    }
}

int main(void)
{
    int n;
    printf("Enter the no of disks\n");
    scanf("%d",&n);
    towersofhanoi(n,'A','B','C');
    return 0;
}
```

OUTPUT:

Output

```
/tmp/rNeJbQ24V1.o
```

```
Enter the no of disks
```

```
3
```

```
move disk no 1 to tower B
```

```
move disk no 2 to tower C
```

```
move disk no 1 to tower C
```

```
move disk no 3 to tower B
```

```
move disk no 1 to tower A
```

```
move disk no 2 to tower B
```

```
move disk no 1 to tower B
```

ALGORITHM:

Algorithm TowersOfHanoi(n,a,b,c)

```
{
```

```
    if  $n \geq 1$  {
```

```
        TowersOfHanoi(n - 1,a,c,b);
```

```
        Print "Move disk no ", n, " to tower ",  
destination
```

```
        TowersOfHanoi(n-1,c,b,a);
```

```
    }
```

```
}
```

BIT STRING:

CODE:

```
#include <stdio.h>

int Binary_Strings(int n) {
    if (n <= 1) {
        return 0;
    } else if (n == 2) {
        return 3;
    } else {
        return Binary_Strings(n - 1) + Binary_Strings(n
- 2);
    }
}

int main() {
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    if (n <= 1) {
        printf("Invalid input\n");
        return 1;
    }
    int result = Binary_Strings(n);
    printf("a%d = %d\n", n, result);
    return 0;
}
```

OUTPUT:

```
Output
/tmp/rNeJbQ24V1.o
Enter the value of n: 5
a5 = 9
|
```

ALGORITHM:

Algorithm Binary_Strings(n)

```
{
    if n ≤ 1 {
        return 0
    }
    else if n = 2 {
        return 3
    }
    else {
        return Binary_Strings(n - 1) + Binary_Strings(n
- 2)
    }
}
```