

NAME : Himanshu Dixit
ENROLL NO. : B64178
BATCH : B10

Software Development Fundamentals – I(15B11CI111)
ODD 2021
Tutorial Sheet – 8

Q1. [CO4] Answer the following (a) and (b) both.

(a) What are actual and formal parameters in call by value functions?

Solution:

| Actual parameters | Formal parameters |
|--|---|
| 1. used in function calling. 2. are the variables constant or expressions contained in a function call that replace the formal parameters which is a part of the function declaration/definition. | 1. used in function declaration. 2. also called dummy arguments. 3. may also replaced by the actual parameters. 4. may be declared by the same name or different name as of actual parameters. |

(b) What are the features of call by value functions?

Solution:

The main feature of call by value that formal argument not affect the value of actual arguments.

Q2. [CO4] Write a program in C to find the sum of the series $1!/1+2!/2+3!/3+4!/4+5!/5$ using the function.

Solution:

```
#include<stdio.h>
int fact(int f);
int main()
{
    float sum=0;
    int n,a;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
    {
        printf("%d!/ %d",i,i);
        if(i!=n)
            printf(" + ");
        a=fact(i);
        sum=sum+(a/i);
    }
    printf("\nsum = %.2f",sum);
    return 0;
```

```

}
int fact(int f)
{
    int i=1,p=1;
    while(i<=f)
    {
        p=p*i;
        i++;
    }
    return p;
}

```

```

1  #include<stdio.h>
2  int fact(int f);
3  int main()
4  {
5      float sum=0;
6      int n,a;
7      scanf("%d",&n);
8      for(int i=1;i<=n;i++)
9      {
10         printf("%d!/ %d",i,i);
11         if(i!=n)
12             printf(" + ");
13         a=fact(i);
14         sum=sum+(a/i);
15     }
16     printf("\nsum = %.2f",sum);
17     return 0;
18 }
19 int fact(int f)
20 {
21     int i=1,p=1;
22     while(i<=f)

```

Output window content:

```

C:\Himanshu864178\patternsum.exe
5
1!/1 + 2!/2 + 3!/3 + 4!/4 + 5!/5
sum = 34.00
Process returned 0 (0x0)   execution time : 1738.750 s
Press any key to continue.

```

Q3. [CO4]What will be the output of the following program?

```

#include<stdio.h>
void modify (int n) {
    n=n+10;
    printf("Value of 'n' inside function=%d \n", n);
}

```

```

int main() {
    int x=15;
    modify(x);//passing value in function
    printf("Modified value of x=%d \n", x);
    return 0;
}

```

Solution:

Value of 'n' inside function=25
Modified value of x=15

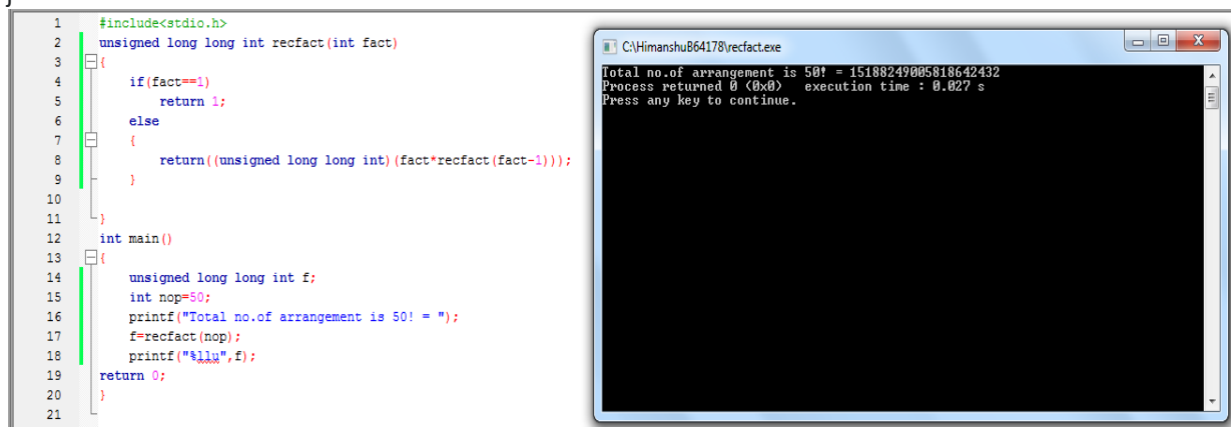
Q4. [CO4]A girl carries 50 hair pins in a haphazard manner within her dressing drawer. Her mother asked her to arrange these pins so that the drawer will look elegant. But the girl is

confused as she does not know the number of possible ways in which she can arrange her hair pins. You must suggest a recursive function to the girl in order to ease her task.

Solution:

```
#include<stdio.h>
unsigned long long int recfact(int fact)
{
    if(fact==1)
        return 1;
    else
    {
        return((unsigned long long int)(fact*recfact(fact-1)));
    }
}

int main()
{
    unsigned long long int f;
    int nop=50;
    printf("Total no.of arrangement is 50! = ");
    f=recfact(nop);
    printf("%llu",f);
    return 0;
}
```

The image shows a code editor on the left and a terminal window on the right. The code editor displays the C program for calculating 50! using a recursive function. The terminal window shows the output of the program, which is the value of 50! (15188249005818642432) and the execution time (0.027 s).

```
1  #include<stdio.h>
2  unsigned long long int recfact(int fact)
3  {
4      if(fact==1)
5          return 1;
6      else
7      {
8          return((unsigned long long int)(fact*recfact(fact-1)));
9      }
10 }
11
12 int main()
13 {
14     unsigned long long int f;
15     int nop=50;
16     printf("Total no.of arrangement is 50! = ");
17     f=recfact(nop);
18     printf("%llu",f);
19     return 0;
20 }
21
```

```
CA\Himanshu864178\recfact.exe
Total no.of arrangement is 50! = 15188249005818642432
Process returned 0 (0x0)   execution time : 0.027 s
Press any key to continue.
```

Q5. [CO4]Write a program in C to convert decimal number to binary number using the function.

Solution:

```
#include<stdio.h>
void conv(int n)
{
    int i;
    int a[10];
    for(i=0;n>0;i++)
    {
        a[i]=n%2;
        n=n/2;
    }
}
```

```

    }
    int k=i-1;
    printf("Binary convert is ");
    for(i=k;i>=0;i--)
        printf("%d",a[i]);
}
int main()
{
    int n;
    scanf("%d",&n);
    conv(n);
return 0;
}

```

```

1  #include<stdio.h>
2  void conv(int n)
3  {
4      int i;
5      int a[10];
6      for(i=0;n>0;i++)
7      {
8          a[i]=n%2;
9          n=n/2;
10     }
11     int k=i-1;
12     printf("Binary convert is ");
13     for(i=k;i>=0;i--)
14         printf("%d",a[i]);
15 }
16 int main()
17 {
18     int n;
19     scanf("%d",&n);
20     conv(n);
21     return 0;
22 }

```

C:\HimanshuB64178\dectobin.exe

```

2
Binary convert is 10
Process returned 0 (0x0)   execution time : 6.728 s
Press any key to continue.

```

Q6. [CO4] Write a program to swap two numbers temporarily using call by value.

Solution:

```

#include<stdio.h>
void swaping(int a,int b)
{
    int k;
    k=b;
    b=a;
    a=k;
    printf("a = %d\nb = %d",a,b);
}
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    swaping(a,b);
    printf("\na = %d\nb = %d",a,b);
return 0;
}

```

```

1  #include<stdio.h>
2  void swapping(int a,int b)
3  {
4      int k;
5      k=b;
6      b=a;
7      a=k;
8      printf("a = %d\nb = %d",a,b);
9  }
10 int main()
11 {
12     int a,b;
13     scanf("%d%d",&a,&b);
14     swapping(a,b);
15     printf("\na = %d\nb = %d",a,b);
16     return 0;
17 }
18

```

```

1 2
a = 2
b = 1
a = 1
b = 2
Process returned 0 (0x0)   execution time : 5.569 s
Press any key to continue.

```

Q7. [CO4] A child is taking his daily lessons on mathematics from online classes. Today he is learning about the counting of stars. He starts counting as one star, two stars, three stars ... then 49 stars and finally 50 stars. Help the child in summing up his counting of the number of stars using recursive function.

Solution:

```

#include<stdio.h>
int recsum(int n,int sum)
{
    if(n==51)
        return sum;
    sum=sum+n;
    n++;
    int s=recsum(n,sum);
    return s;
}
int main()
{
    int n=1,sum=0;
    printf("sum upto 50\n");
    int s=recsum(n,sum);
    printf("%d",s);
    return 0;
}

```

```

1  #include<stdio.h>
2  int recsum(int n,int sum)
3  {
4      if(n==51)
5          return sum;
6      sum=sum+n;
7      n++;
8      int s=recsum(n,sum);
9      return s;
10 }
11 int main()
12 {
13     int n=1,sum=0;
14     printf("sum upto 50\n");
15     int s=recsum(n,sum);
16     printf("%d",s);
17     return 0;
18 }
19

```

```

sum upto 50
1275
Process returned 0 (0x0)   execution time : 0.023 s
Press any key to continue.

```

Q8. [CO4] Write a program in C to check armstrong and perfect numbers using the function.

Solution:

```
#include<stdio.h>
void armstrong(int a)
{
    int n,num=a,sum=0;
    while(a>0)
    {
        n=a%10;
        a=a/10;
        sum=sum+n*n*n;
    }
    if(num==sum)
        printf("%d is armstrong number\n",num);
    else
        printf("%d is not armstrong number\n",num);
}
void perfect(int a)
{
    int i=1,sum=0,num=a;
    while(i<=a)
    {
        if(a%i==0)
            sum=sum+i;
        i++;
    }
    if(num*2==sum)
        printf("%d is perfect number\n",num);
    else
        printf("%d is not perfect number\n",num);
}
int main()
{
    int a;
    scanf("%d",&a);
    armstrong(a);
    perfect(a);
    return 0;
}
```

```
1  #include<stdio.h>
2  void armstrong(int a)
3  {
4      int n,num=a,sum=0;
5      while(a>0)
6      {
7          n=a%10;
8          a=a/10;
9          sum=sum+a*a*a;
10     }
11     if(num==sum)
12         printf("%d is armstrong number\n",num);
13     else
14         printf("%d is not armstrong number\n",num);
15 }
16 void perfect(int a)
17 {
18     int i=1,sum=0,num=a;
19     while(i<=a)
20     {
21         if(a%i==0)
22             sum=sum+i;
```

