

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

SOFTWARE DEVELOPMENT FUNDAMENTAL LAB-I(15B17CI171)
Assignment Sheet (WEEK-7 PHASE-2)
Lab A

- 1. Write a function in c to find the missing value in an array. For example:**
If array A= { 1, 2, 4, 6, 3, 7, 8} Missing value is 5
If A={1,2,4,5} Missing value is 3.

Solution:

```
#include<stdio.h>
void missing(int a[100],int n)
{
    int k=0;
    for(int i=a[0];i<=a[n-1];i++)
    {
        if(i!=a[k])
        {
            printf("%d is missing number\n",i);
            k--;
        }
        k++;
    }
}
int main()
{
    int a[100],n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    missing(a,n);
    return 0;
}
```

```

1  #include<stdio.h>
2  void missing(int a[100],int n)
3  {
4      int k=0;
5      for(int i=a[0];i<=a[n-1];i++)
6      {
7          if(i!=a[k])
8          {
9              printf("%d is missing number\n",i);
10             k--;
11         }
12         k++;
13     }
14 }
15 int main()
16 {
17     int a[100],n;
18     scanf("%d",&n);
19     for(int i=0;i<n;i++)
20     {
21         scanf("%d",&a[i]);
22     }

```

```

C:\HimanshuB64178\missing.exe
6
1 2 3 6 7 8
4 is missing number
5 is missing number
Process returned 0 (0x0) execution time : 183.523 s
Press any key to continue.

```

2. Write a C/C++ function “array_sort” which arranges the numbers of an array in ascending order. Consider that the user will enter the number randomly for array in the main function.

Solution:

```

#include<stdio.h>
void arraysort(int a[100],int n)
{
    int k;
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                k=a[j];
                a[j]=a[i];
                a[i]=k;
            }
        }
    }
    for(int i=0;i<n;i++)
        printf("%d ",a[i]);
}
int main()
{
    int a[100],n;

```

```

scanf("%d",&n);
for(int i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
arraysort(a,n);
return 0;
}

```

```

1  #include<stdio.h>
2  void arraysort(int a[100],int n)
3  {
4      int k;
5      for(int i=0;i<n;i++)
6      {
7          for(int j=i+1;j<n;j++)
8          {
9              if(a[i]>a[j])
10             {
11                 k=a[j];
12                 a[j]=a[i];
13                 a[i]=k;
14             }
15         }
16     }
17     for(int i=0;i<n;i++)
18         printf("%d ",a[i]);
19 }
20 int main()
21 {
22     int a[100],n;

```

```

C:\Himanshu864178\sortarray.exe
4 7 2 1 8 9
1 2 4 7 8 9
Process returned 0 (0x0)   execution time : 12.712 s
Press any key to continue.

```

3. Write a program to find sum of digits of the number using Recursive Function.

Solution:

```

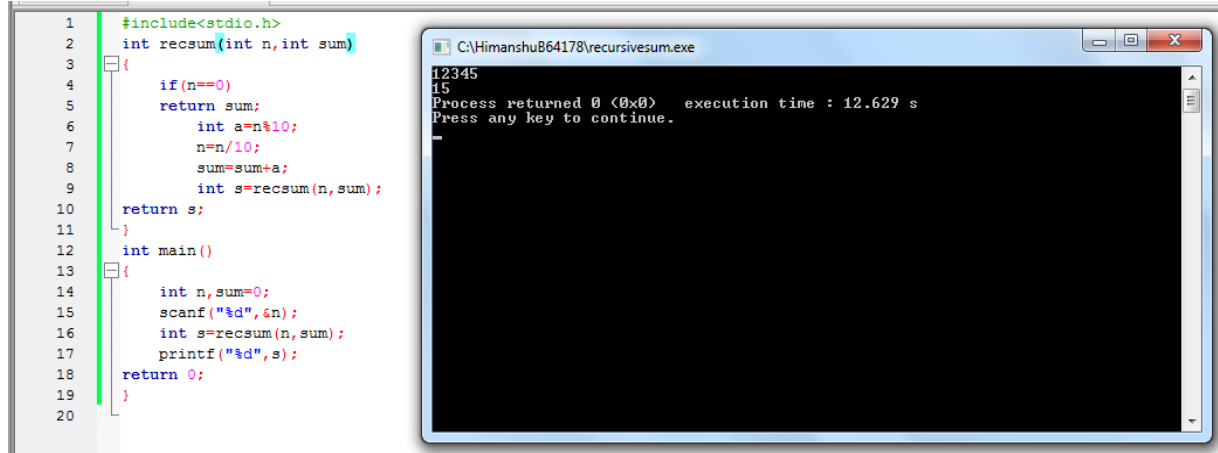
#include<stdio.h>
int recsum(int n,int sum)
{
    if(n==0)
        return sum;
    int a=n%10;
    n=n/10;
    sum=sum+a;
    int s=recsum(n,sum);
    return s;
}
int main()
{
    int n,sum=0;
    scanf("%d",&n);

```

```

    int s=recsum(n,sum);
    printf("%d",s);
return 0;
}

```



The screenshot shows a C++ IDE with a file named `recsum.cpp` and its execution output in a separate window.

Code in `recsum.cpp`:

```

1  #include<stdio.h>
2  int recsum(int n,int sum)
3  {
4      if(n==0)
5          return sum;
6      int a=n%10;
7      n=n/10;
8      sum=sum+a;
9      int s=recsum(n,sum);
10     return s;
11 }
12 int main()
13 {
14     int n,sum=0;
15     scanf("%d",&n);
16     int s=recsum(n,sum);
17     printf("%d",s);
18     return 0;
19 }
20

```

Execution Output:

```

C:\Himanshu864178\recsum.exe
12345
15
Process returned 0 (0x0)   execution time : 12.629 s
Press any key to continue.

```

4. Write a program to read an integer number and print the reverse of that number using recursion.

Solution:

```

#include<stdio.h>
int recrev(int n,int rev)
{
    if(n==0)
        return rev;
    int a=n%10;
    n=n/10;
    rev=rev*10+a;
    int r=recrev(n,rev);
return r;
}
int main()
{
    int n,rev=0;
    scanf("%d",&n);
    int r=recrev(n,rev);
    printf("%d",r);
return 0;
}

```

```
1  #include<stdio.h>
2  int recrev(int n,int rev)
3  {
4      if(n==0)
5          return rev;
6          int a=n%10;
7          n=n/10;
8          rev=rev*10+a;
9          int r=recrev(n,rev);
10     return r;
11 }
12 int main()
13 {
14     int n,rev=0;
15     scanf("%d",&n);
16     int r=recrev(n,rev);
17     printf("%d",r);
18     return 0;
19 }
20
```

C:\Himanshu864178\reverserec.exe

12345
54321
Process returned 0 (0x0) execution time : 3.372 s
Press any key to continue.