LAB VII

1. Run the following program and examine the output.

```
#include <stdio.h>
int main()
{
int a=5,*b,**c=&b;
b=&a;
*b=7;
printf("%d   %d  %d\n",a,*b,**c);
printf("%p   %p  %p\n",&a,&b,&c);
printf("%p  %p\n",b,*c);
printf("%p\n",c);

return 0;
}
```

2. Run the following program and examine the output.

```
#include <stdio.h>
int sum1(int,int);
int sum2(int*,int*);
void sum3(int*,int*,int*);
int main()
{
int a,b,s;
a=5,b=7;
printf("Before calling sum1: a=%d  b=%d\n",a,b);
s=sum1(a,b);
printf("After calling sum1: a=%d  b=%d  s=%d\n",a,b,s);

a=8,b=9;
printf("Before calling sum2: a=%d  b=%d\n",a,b);
s=sum2(&a,&b);
printf("After calling sum2: a=%d  b=%d  s=%d\n",a,b,s);

a=3,b=4;
printf("Before calling sum3: a=%d  b=%d\n",a,b);
sum3(&a,&b,&s);
printf("After calling sum3: a=%d  b=%d  s=%d\n",a,b,s);

return 0;
}
int sum1(int a,int b)
{
int sum,t;
sum=a+b;
t=a;
a=b;
b=t;
return sum;
}

int sum2(int *a,int *b)
{
int sum,t;
sum=*a+*b;
t=*a;
*a=*b;
*b=t;
return sum;
}

void sum3(int *a,int *b,int *s)
{
int t;
*s=*a+*b;
t=*a;
*a=*b;
*b=t;
}
```

3. Complete the given code (omitting the comments) with modifications at appropriate places (see the comment parts of the program). The program computes the value of

$$\tanh^{-1} x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1}, \qquad |x| < 1$$

using a **void** function **valitanh**. After call to the function, integer variable flag will have value **0** if x is outside the range. Otherwise, flag will have value **1** and val will have value of $\tanh^{-1} x$ using **20** terms of the Taylor series. (*Do not* use **pow** function).

```
#include<stdio.h>
//Write the function prototype here
int main()
{
int flag,nterms=20;
double x,val;
//Read the value of x from the keyboard
valitanh(x,&val,nterms,&flag);
if(flag==0)
  printf("x is outside the range\n");
else
  printf("Value is %0.3lf\n",val);
return 0;
}
//Write the function here
```

Test data and expected output:

```
Enter the value of x :-2
x is outside the range

Enter the value of x :0.5
Value is 0.5493

Enter the value of x :-0.2
Value is -0.2027
```

4. Copy the given code (omitting the comments) with modifications at appropriate places (see the comment parts of the program). After completion of the function **vecread**, m and n contain the actual dimensions of u and v and the respective components are stored in u and v. These are read from the files vecu.dat and vecv.dat. The function **vecprint** prints the respective vectors in the terminal. After completion of the function **calc**, *lu* and *lv* contain the length of the vectors. Further, if flag=1, then angle contains the angle between u and v; otherwise flag=0. After completion of the function **writeoutp**, length of the vectors will be written on output.dat. This file will also contain the angle between the vectors if flag=1. In each input data file, the first line contains the dimension of the vector and the next line contains the components of the vector.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 100
//Write the prototypes of the functions vecread,vecprint,calc,writeoutp  here
int main()
{
int m,n,flag;
double u[N],v[N],lu,lv,angle;

vecread(u,"vecu.dat",&m);
vecprint(u,'u',m);
vecread(v,"vecv.dat",&n);
vecprint(v,'v',n);
flag=calc(u,v,m,n,&lu,&lv,&angle);
writeoutp("output.dat",lu,lv,angle,flag);
return 0;
}

//Write details of function calc here

//Write details of function writeup here

//Write details of function vecread here

//Write details of function vecprint here
```

Expected input:
For vecu.dat

```
5
3.0  -4.0  7.0  8.0  9.0
```

and vecv.dat

```
5
4.0  -5.0  7.0  12.0  9.0
```

Expected output:

```
The vector u is :3.0000  -4.0000  7.0000  8.0000  9.0000
The vector v is :4.0000  -5.0000  7.0000  12.0000  9.0000
```

Content of output.dat

```
Length of u is 14.7986
Length of v is 17.7482
The angle between u and v is 10.7973  degrees
```

Expected input:
For vecu.dat

```
6
3.0  -4.0  7.0  8.0  9.0 5.0
```

and vecv.dat

```
5
4.0  -5.0  7.0  12.0  9.0
```

Expected output:

```
The vector u is :3.0000  -4.0000  7.0000  8.0000  9.0000  5.0000
The vector v is :4.0000  -5.0000  7.0000  12.0000  9.0000
```

Content of output.dat

```
Length of u is 15.6205
Length of v is 17.7482
```