MA144: Problem Solving and Computer Programming

Lecture-20

Functions-3

(call by reference, library functions)

Tower of Hanoi Puzzle

```
#include<iostream>
using namespace std;
void TOH(int,char,char,char);
int main()
{   int n;
    cout<<" enter number of disks: ";
    cin>>n;
    TOH(n,'1','2','3');
    return 0;
}
void TOH(int n,char peg1,char peg2,char peg3)
{
    if(n==1)
    { cout<<"\n move top disk from peg "<<peg1<<" to "<<"peg "<<peg2;
      return;
    }
    TOH(n-1,peg1,peg3,peg2);
    cout<<"\n move top disk from peg "<<peg1<<" to "<<"peg "<<peg2;
    TOH(n-1,peg3,peg2,peg1);
 }
```

```
enter number of disks: 3

move top disk from peg 1 to peg 2
move top disk from peg 1 to peg 3
move top disk from peg 2 to peg 3
move top disk from peg 1 to peg 2
move top disk from peg 3 to peg 1
move top disk from peg 3 to peg 2
move top disk from peg 1 to peg 2
```

Find out the output of the program.

```
#include <iostream>
using namespace std;
void disp(int);
int main()
{    int m;
     cout<<"enter an integer: ";
     cin>>m;
     disp(m);
     return 0;
}
void disp(int n)
{  if (n==0)
     return;
   cout<<n<<' ';
   disp(--n);
   }
```

```
#include<iostream>
using namespace std;

int main()
{  int m, i;
   cout<<"enter an integer: ";
   cin>>m;

   for(i=m;i>0;i--)
      cout<<i<<' ';
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
void disp(int);
int main()
{    int m;
     cout<<"enter an integer: ";
     cin>>m;
     disp(m);
     return 0;
}
void disp(int n)
{  if (n==0)
     return;
   cout<<n<<' ';
   disp(--n);
   }
```

```
enter an integer: 12
12 11 10 9 8 7 6 5 4 3 2 1
```

End of file using do-while loop

```
#include <iostream>
using namespace std;
void disp(int);
int main()
{
    char ch;
    int m;

    do{
    cout<<"enter an integer: ";
    cin>>m;
    disp(m);
    cout << "\n Do you want to Continue? Enter your choice (y / n):";
    cin >> ch;
    }while( ch == 'y' || ch == 'Y');

    return 0;
}
void disp(int n)
{  if (n==0)
    return;
  cout<<n<<' ';
  disp(--n);
  }
```

```
enter an integer: 10
10 9 8 7 6 5 4 3 2 1
 Do you want to Continue? Enter your choice (y / n):y
enter an integer: 6
6 5 4 3 2 1
 Do you want to Continue? Enter your choice (y / n):n
```

Call by reference

Call by reference

When we pass arguments by reference, the formal arguments in the **called function** become aliases to the actual arguments in the **calling function**.

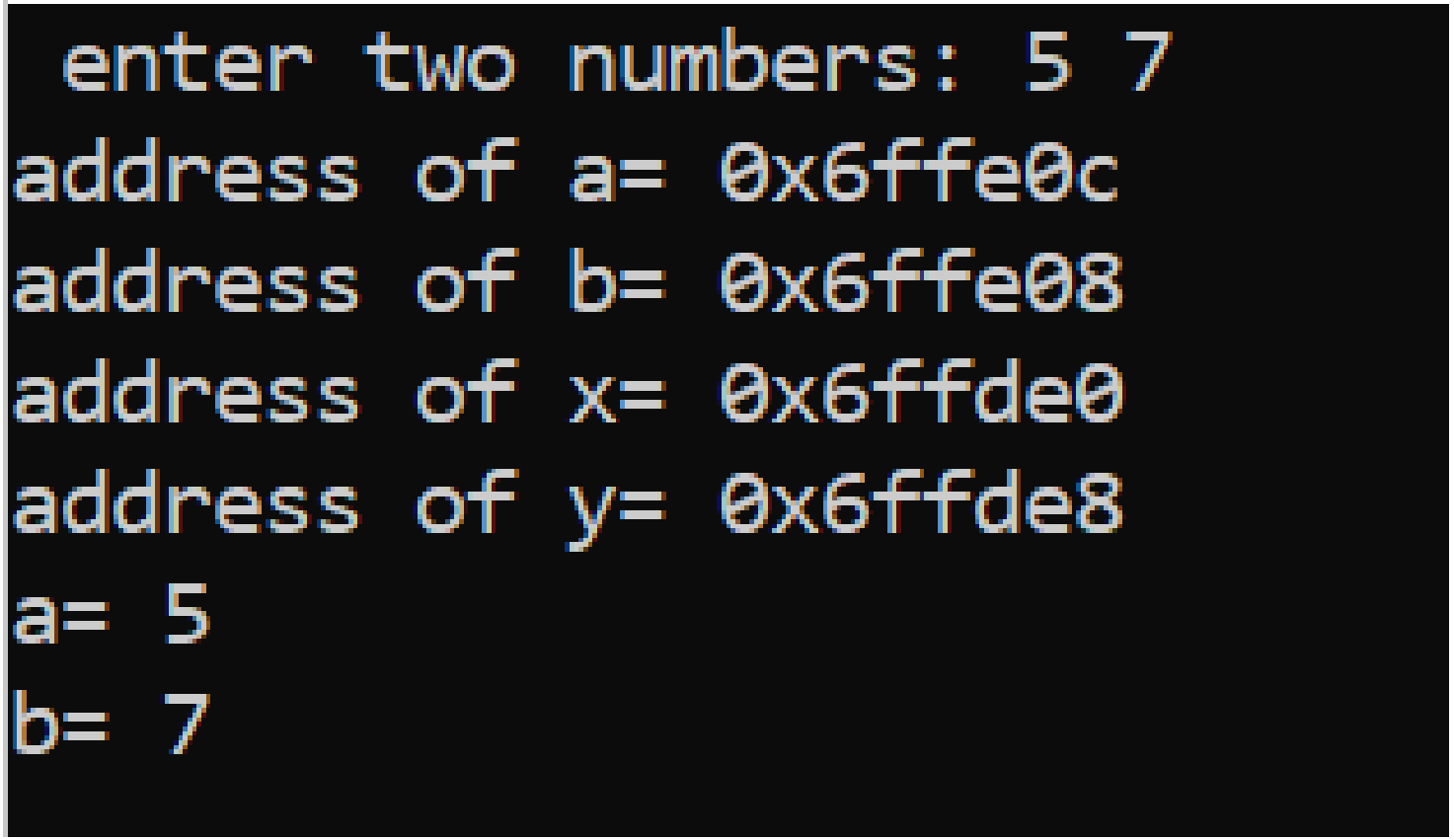
Call-by-value program

```
#include<iostream>
using namespace std;
void call_by_val(int,int);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     call_by_val(a,b);
     cout<<"a= "<<a<<endl<<"b= "<<b;
          return 0;
}
void call_by_val(int x,int y)
{ x+=2;  y+=5;
 }
```

```
 enter two numbers: 5 10
a= 5
b= 10
```

Actual and formal arguments having the different addresses in call by value function call (different variable names)

```
#include<iostream>
using namespace std;
void call_by_val(int,int);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     cout<<"address of a= "<<&a<<endl<<"address of b= "<<&b<<endl;
     call_by_val(a,b);
     cout<<"a= "<<a<<endl<<"b= "<<b<<endl;
     return 0;
}
void call_by_val(int x,int y)
{ x+=2;  y+=5;
cout<<"address of x= "<<&x<<endl<<"address of y= "<<&y<<endl;
 }
```


enter two numbers: 5 7
address of a= 0x6ffe0c
address of b= 0x6ffe08
address of x= 0x6ffde0
address of y= 0x6ffde8
a= 5
b= 7

Actual and formal arguments having the different addresses in call by value function call (same variable names)

```
#include<iostream>
using namespace std;
void call_by_val(int,int);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     cout<<"address of actual a= "<<&a<<endl;
     cout<<"address of actual b= "<<&b<<endl;
     call_by_val(a,b);
     cout<<"a= "<<a<<endl<<"b= "<<b<<endl;
     return 0;
}
void call_by_val(int a,int b)
{ a+=2;  a+=5;
cout<<"address of formal a= "<<&a<<endl;
cout<<"address of formal b= "<<&b<<endl;
 }
```

```
 enter two numbers: 5 7
address of actual a= 0x6ffe0c
address of actual b= 0x6ffe08
address of formal a= 0x6ffde0
address of formal b= 0x6ffde8
a= 5
b= 7
```

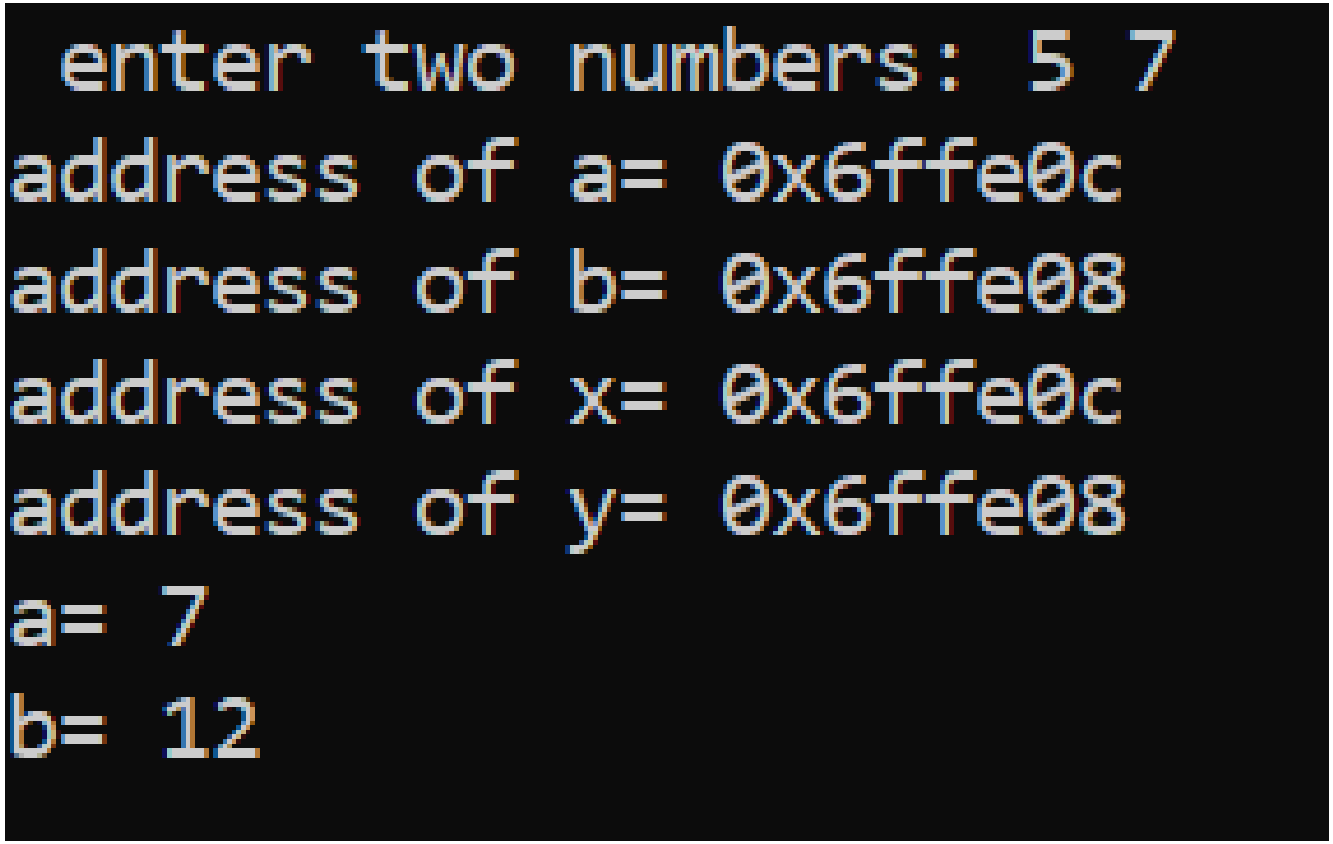
Call-by-reference program

```
#include<iostream>
using namespace std;
void call_by_val(int&,int&);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     call_by_val(a,b);
     cout<<"a= "<<a<<endl<<"b= "<<b;
          return 0;
}
void call_by_val(int& x,int& y)
{ x+=2;  y+=5;
 }
```

```
 enter two numbers: 5 10
a= 7
b= 15
```

Actual and formal arguments having the same addresses in call by reference function call (different variable names)

```
#include<iostream>
using namespace std;
void call_by_val(int&,int&);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     cout<<"address of a= "<<&a<<endl<<"address of b= "<<&b<<endl;
     call_by_val(a,b);
     cout<<"a= "<<a<<endl<<"b= "<<b<<endl;
     return 0;
}
void call_by_val(int& x,int& y)
{ x+=2;  y+=5;
cout<<"address of x= "<<&x<<endl<<"address of y= "<<&y<<endl;
 }
```


enter two numbers: 5 7
address of a= 0x6ffe0c
address of b= 0x6ffe08
address of x= 0x6ffe0c
address of y= 0x6ffe08
a= 7
b= 12

Actual and formal arguments having the same addresses in call by reference function call (same variable names)

```
#include<iostream>
using namespace std;
void call_by_val(int&,int&);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     cout<<"address of actual a= "<<&a<<endl;
     cout<<"address of actual b= "<<&b<<endl;
     call_by_val(a,b);
     cout<<"a= "<<a<<endl<<"b= "<<b<<endl;
     return 0;
}
void call_by_val(int& a,int& b)
{ a+=2;  a+=5;
cout<<"address of formal a= "<<&a<<endl;
cout<<"address of formal b= "<<&b<<endl;
 }
```

```
 enter two numbers: 5 7
address of actual a= 0x6ffe0c
address of actual b= 0x6ffe08
address of formal a= 0x6ffe0c
address of formal b= 0x6ffe08
a= 12
b= 7
```

Swapping two numbers

```
#include<iostream>
using namespace std;
void swap(int&,int&);
int main()
{    int a,b;
     cout<<" enter two numbers: ";
     cin>>a>>b;
     cout<<"a= "<<a<<endl<<"b= "<<b<<endl;
     swap(a,b);
     cout<<"after swapping "<<"\n a= "<<a<<endl<<"b= "<<b;
          return 0;
}
void swap(int& x,int& y)
{ int t;
  t=x;
  x=y;
  y=t;
 }
```

```
 enter two numbers: 5 10
a= 5
b= 10
after swapping
 a= 10
b= 5
```

Library functions

Arithmetic Functions

Function Declaration	Description	Header File
`int abs(int);`	Absolute value	`cstdlib`
`long labs(long);`	Absolute value	`cstdlib`
`double fabs(double);`	Absolute value	`cmath`
`double sqrt(double);`	Square root	`cmath`
`double pow(double, double);`	Returns the first argument raised to the power of the second argument.	`cmath`
`double exp(double);`	Returns e (base of the natural logarithm) to the power of its argument.	`cmath`
`double log(double);`	Natural logarithm (ln)	`cmath`
`double log10(double);`	Base 10 logarithm	`cmath`
`double ceil(double);`	Returns the smallest integer that is greater than or equal to its argument.	`cmath`
`double floor(double);`	Returns the largest integer that is less than or equal to its argument.	`cmath`

Trigonometric Functions

These functions use radians, not degrees.

Function Declaration	Description	Header File
`double acos(double);`	Arc cosine	`cmath`
`double asin(double);`	Arc sine	`cmath`
`double atan(double);`	Arc tangent	`cmath`
`double cos(double);`	Cosine	`cmath`
`double cosh(double);`	Hyperbolic cosine	`cmath`
`double sin(double);`	Sine	`cmath`
`double sinh(double);`	Hyperbolic sine	`cmath`
`double tan(double);`	Tangent	`cmath`
`double tanh(double);`	Hyperbolic tangent	`cmath`

```cpp
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
cout<<cos(5.6);
cout<<endl;
cout<<sqrt(25);
cout<<endl;
cout<<fabs(-2.5);
cout<<endl;
cout<<ceil(2.8);
cout<<endl;
cout<<ceil(-2.8);
return 0;
}
```

```
0.775566
5
2.5
3
-2
```