Data Structures and Algorithms – Lab 6

Iulia-Cristina Stanica
iulia.stanica@gmail.com

Roadmap

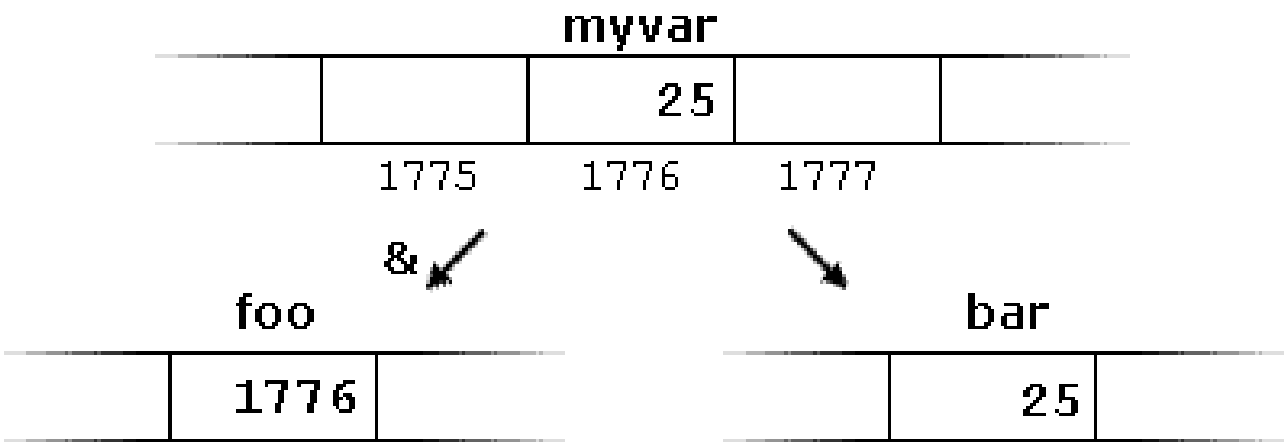
- Pointers
- Pointers and functions
- Pointers and arrays
- Pointers to pointers

Pointers

- A pointer = variable containing **the address** of a different variable (reference to another variable)
- Pointers save an address, so we should use an appropriate operator <u>(we use &)</u>

Pointers

```
myvar = 25;
foo = &myvar;
bar = myvar;
```



Pointers

- Declaring a pointer to a variable of type int / double:
  - int *px; or int* px;
  - double *py; or double* py;

  - int xVal = 5;
  - int *px = &xVal; //the address of xVal is saved in the pointer

- For getting the **content** of the "pointed" variable, we use the dereference <u>(we use *):</u>
  - int cont = *px;

Example 1

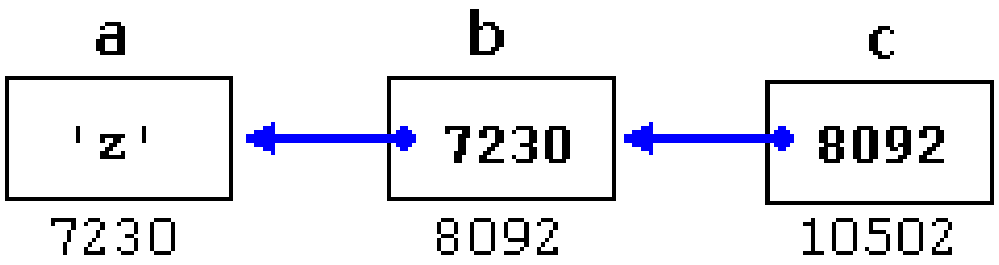
```
#include <iostream>
using namespace std;

int main () {
   int *px; //declare pointer
   int x = 5; //declare int variable
   px = &x; //assign address of the variable to the pointer

   cout<<"Address of x: "<<&x<<endl;
   cout<<“Value of x: "<<x<<endl;
   cout<<"Value of pointer px: "<<px<<endl;
   cout<<"Px points to: "<<*px<endl;
   // & - reference, * - dereference
   return 0;
}
```

Pointers

```
1 char a;
2 char * b;
3 char ** c;
4 a = 'z';
5 b = &a;
6 c = &b;
```



Example 2

```
// more pointers
#include <iostream>
using namespace std;

int main ()
{
  int firstvalue = 5, secondvalue = 15;
  int * p1, * p2;

  p1 = &firstvalue;  // p1 = address of firstvalue
  p2 = &secondvalue; // p2 = address of secondvalue
  *p1 = 10;          // value pointed to by p1 = 10
  *p2 = *p1;         // value pointed to by p2 = value pointed
    to by p1
  p1 = p2;           // p1 = p2 (value of pointer is copied)
  *p1 = 20;          // value pointed to by p1 = 20

  cout << "firstvalue is " << firstvalue << '\n';
  cout << "secondvalue is " << secondvalue << '\n';
  return 0;
}
```

What is the result?

Example 3 – pointers and functions

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

int swap1(int a, int b) //parameters passed by value
   {
         int x = a;
         a = b;
         b = x;
   }
int swap2(int &a, int &b) //parameters passed by address
   {
         int x = a;
         a = b;
         b = x;
   }

int swap3(int *a, int *b) //the parameters are pointers, we
   must call the function using addresses
   {
         int x = *a;
         *a = *b;
         *b = x;
   }
```

```
int main()
{
      int a = 15;
      int b = 38;

      int *pa;
      pa = &a;

      int *pb;
      pb = &b;

      swap1(a,b);
      cout<<a<<"|"<<b<<"\n";

      swap2(a,b);
      cout<<a<<"|"<<b<<"\n";

      swap3(&a,&b); //call using
      addresses
      cout<<a<<"|"<<b<<"\n";

      swap3 (pa,pb);
      cout<<a<<"|"<<b<<"\n";

      return 0;
}
```

Example 4 – pointers and char arrays

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

int main()
{
char s[] = "Hello world"; //array of chars

char *ps;
ps = &s[0]; //ps – pointer to the first element of the string

char *pa;
pa = s; //same thing, pointer to the first element of the string

cout<<"ps points at:"<<*ps<<endl; //dereference, we get the first letter

//Special case: we display a sequence of chars instead of an address (the operator << gets char*
    as a string (function overloading))
cout<<ps<<endl;

cout<<(void *)ps<<endl; //in reality, the pointer contains the address (forced conversion)

while (*ps) //if the pointer references a value…
{
    cout<<"L'adresse: "<<(void *) ps<<" et le contenu: "<<*ps<<endl;
    ps++; //we pass to the next address
}
return 0; }
```

Example 5 – pointers and matrix

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

int main() {
    int mat[3][3] = {{2,3,4}, {5,6,7}, {8,9,10}};
    int *pmat = mat[0]; //mat[0] contains the address
                        //of the first element of the matrix

 // int *pmat = &mat[0][0]; – what is the effect of this?

    for (int i=0; i<9; i++)
        cout<<*(pmat++) << " ";


return 0;
}
```

Example 6 – pointers, functions and arrays

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

void modify1 (int a[])
{
    a[2]=15;
}

void modify2 (int *a)
{
    *(a+2) = 15; //same effect as
    modify1
}
```

```
int main()
{
    int x[]={1,2,3,4};
    cout<<"Before the fonction:" <<
x[2] << endl;
    modify1(x);

    int *p=x; //same:
    //int *p = &x[0];
    // modify2(p);
    // modify2(x);
    cout<<"After the function: " << x[2]
<< endl;

return 0;
}
```

Example 7 – Pointers to pointers

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

int main()
{
    string day[] = {“Moday", “Tuesday", “Wednesday"};
    string *pday = day;  //string *pday = &day[0]; -> same
    string **ppday = &pday; //pointer to pointer

    cout<<“A: Value: "<<*pday<<", at the address: "<<pday<<endl;

    cout<<“B: Value: "<<**ppday<<", to the address of the level 1 pointer (pday): “
       << *ppday <<endl;
    cout<< " and the address of the level 2 pointer (ppday): "<< ppday<<endl;

    if (*ppday==pday)
       cout<<“Same addresses!";

    return 0;
}
```

Exercises

```
int *p;
int i;
int k;
i = 42;
k = i;
p = &i;
```

After which instruction, i becomes 75?

A. k = 75;

B. *k = 75;

C. p = 75;

D. *p = 75;

E. Multiple correct answers.

Ex 1. Write a program taking 6 values which are saved in an array with the help of a pointer. Display on the screen the elements of the array.

Ex 2. Print the elements of the array in the reverse order with the help of a pointer.

- Ex 3. Reverse the array using only pointers.
  Hint! Write this function having as parameters a pointer to the array and the size of the array.

Ex: a = {12, 3, 4, 6, 10, 15} becomes: a = {15, 10, 6, 4, 3, 12}

Ex. 4. Write a program to extract the department from a char sequence. Use a function of type:

char *getDep (char *p)

which returns a pointer to the first letter of the department. The p parameter is a pointer to the char sequence.

Ex: We have: “Slatina, OT” and we display using the pointer: “The department is: OT”.

Ex. 5. Write a program to replace commas with blank spaces. Use a function of type:

char *commaReplacer (char *p)

Ex: We have: “College,of,London,” and we get: “College of London ”.