Data Structures 2012-2013 Fall

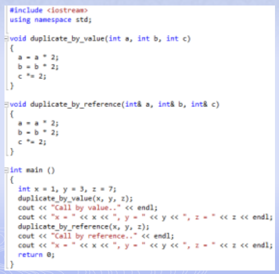
Practice Session 1
Res. Asst. Doğan Altan
Res. Asst. Figen Öztürk
16.10.2012

Contents

Function Calls

- There are 2 ways for passing a parameter to a function in C++.
  1. Call by Value
  2. Call by Reference

Function Calls



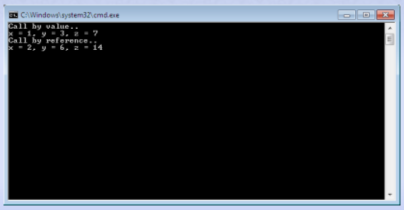
```cpp
#include <iostream>
using namespace std;

void duplicate_by_value(int a, int b, int c)
{
  a = a * 2;
  b = b * 2;
  c *= 2;
}

void duplicate_by_reference(int& a, int& b, int& c)
{
  a = a * 2;
  b = b * 2;
  c *= 2;
}

int main ()
{
  int x = 1, y = 3, z = 7;
  duplicate_by_value(x, y, z);
  cout << "Call by value.." << endl;
  cout << "x = " << x << ", y = " << y << ", z = " << z << endl;
  duplicate_by_reference(x, y, z);
  cout << "Call by reference.." << endl;
  cout << "x = " << x << ", y = " << y << ", z = " << z << endl;
  return 0;
}
```

Function Calls

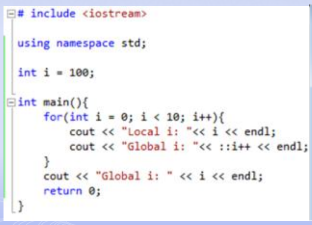


```
Call by value..
x = 1, y = 3, z = 7
Call by reference..
x = 2, y = 6, z = 14
```

Scope

- A variable can be accessed *directly* only in the block it is defined.
- Scope Resolution Operator
  - ::

Scope Example

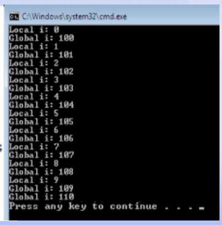


```
# include <iostream>

using namespace std;

int i = 100;

int main(){
    for(int i = 0; i < 10; i++){
        cout << "Local i: "<< i << endl;
        cout << "Global i: "<< ::i++ << endl;
    }
    cout << "Global i: " << i << endl;
    return 0;
}
```

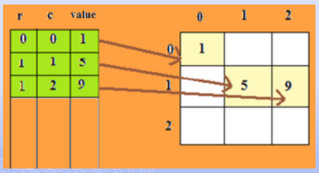


C:\Windows\system32\cmd.exe

```
Local i: 0
Global i: 100
Local i: 1
Global i: 101
Local i: 2
Global i: 102
Local i: 3
Global i: 103
Local i: 4
Global i: 104
Local i: 5
Global i: 105
Local i: 6
Global i: 106
Local i: 7
Global i: 107
Local i: 8
Global i: 108
Local i: 9
Global i: 109
Global i: 110
Press any key to continue . . .
```

Sparse Matrix Application

- A **sparse matrix** is a matrix populated primarily with zeros and only a few of elements are different from zero.

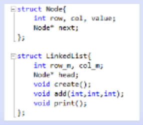


Sparse Matrix Application

- In this application,
  - Row, column and value of non-zero elements are entered from the keyboard.
  - This values are stored in a sorted linked list.
  - While printing out the matrix on the screen, empty elements are printed as ‘0’.

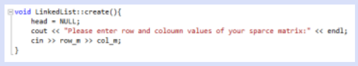
Creating a Sparse Matrix

- The operations for taking row, column and value of non-zero elements of the first matrix from the user.



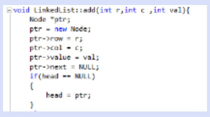
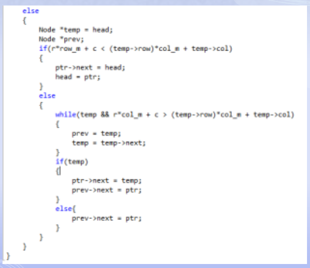
```
struct Node{
    int row, col, value;
    Node* next;
};

struct LinkedList{
    int row_m, col_m;
    Node* head;
    void create();
    void add(int,int,int);
    void print();
};
```



```
void LinkedList::create(){
    head = NULL;
    cout << "Please enter row and coloumn values of your sparce matrix:" << endl;
    cin >> row_m >> col_m;
}
```

Creating Linked List

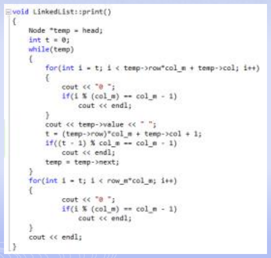



```
void LinkedList::add(int r,int c ,int val){
    Node *ptr;
    ptr = new Node;
    ptr->row = r;
    ptr->col = c;
    ptr->value = val;
    ptr->next = NULL;
    if(head == NULL)
    {
        head = ptr;
    }
```

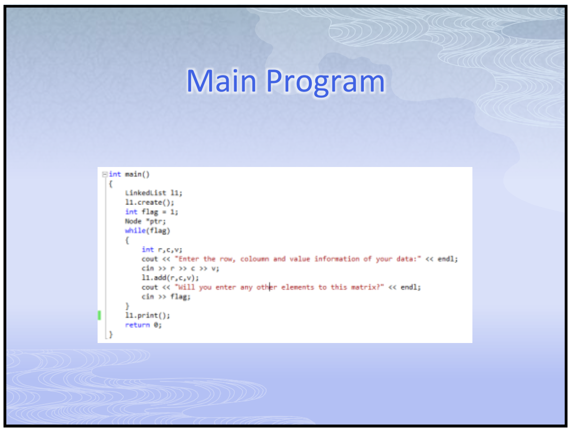
```
    else
    {
        Node *temp = head;
        Node *prev;
        if(r*row_m + c < (temp->row)*col_m + temp->col)
        {
            ptr->next = head;
            head = ptr;
        }
        else
        {
            while(temp && r*col_m + c > (temp->row)*col_m + temp->col)
            {
                prev = temp;
                temp = temp->next;
            }
            if(temp)
            {
                ptr->next = temp;
                prev->next = ptr;
            }
            else{
                prev->next = ptr;
            }
        }
    }
}
```

Printing out on the Screen

- Printing out the matrix in “print()” method

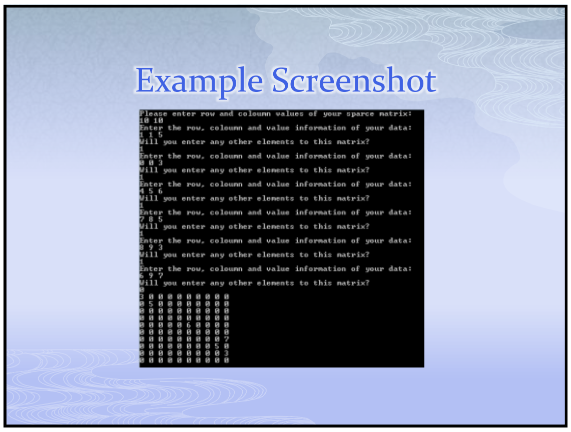


```
void LinkedList::print()
{
    Node *temp = head;
    int t = 0;
    while(temp)
    {
        for(int i = t; i < temp->row*col_m + temp->col; i++)
        {
            cout << "0 ";
            if(i % (col_m) == col_m - 1)
                cout << endl;
        }
        cout << temp->value << " ";
        t = (temp->row)*col_m + temp->col + 1;
        if((t - 1) % col_m == col_m - 1)
            cout << endl;
        temp = temp->next;
    }
    for(int i = t; i < row_m*col_m; i++)
    {
            cout << "0 ";
            if(i % (col_m) == col_m - 1)
                cout << endl;
    }
    cout << endl;
}
```



Main Program

```
int main()
{
    LinkedList l1;
    l1.create();
    int flag = 1;
    Node *ptr;
    while(flag)
    {
        int r,c,v;
        cout << "Enter the row, coloumn and value information of your data:" << endl;
        cin >> r >> c >> v;
        l1.add(r,c,v);
        cout << "Will you enter any other elements to this matrix?" << endl;
        cin >> flag;
    }
    l1.print();
    return 0;
}
```



Example Screenshot

```
Please enter row and coloumn values of your sparce matrix:
10 10
Enter the row, coloumn and value information of your data:
1 1 5
Will you enter any other elements to this matrix?
1
Enter the row, coloumn and value information of your data:
0 0 3
Will you enter any other elements to this matrix?
1
Enter the row, coloumn and value information of your data:
4 5 6
Will you enter any other elements to this matrix?
1
Enter the row, coloumn and value information of your data:
7 8 5
Will you enter any other elements to this matrix?
1
Enter the row, coloumn and value information of your data:
8 9 3
Will you enter any other elements to this matrix?
1
Enter the row, coloumn and value information of your data:
6 9 7
Will you enter any other elements to this matrix?
0
3 0 0 0 0 0 0 0 0 0
0 5 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 6 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 7
0 0 0 0 0 0 0 0 5 0
0 0 0 0 0 0 0 0 0 3
0 0 0 0 0 0 0 0 0 0
```