Lecture 23 Pointers, LinkedList

1. Pointers
2. #include <iostream>
3. using namespace std;
4. int main() {
5. // int x = 10;
6. // int y = 20;
7. // int \*xptr;
8. // xptr = &x;
9. // cout<<&x<<endl;
10. // cout<<xptr<<endl;
11. // xptr = &y;
12. // cout<<&y<<endl;
13. // cout<<xptr<<endl;
14. int x = 10;
15. int \*xptr;
16. xptr = &x;
17. cout<<&x<<endl;
18. cout<<xptr<<endl;
19. cout<<\*(&x)<<endl;
20. cout<<\*(xptr)<<endl;
21. cout<<\*(&xptr)<<endl;
22. cout<<&(\*xptr)<<endl;
23. cout<<&xptr<<endl;
24. int \*\*xxptr = &xptr;
25. cout<<xxptr<<endl;
26. return 0;
27. }

2. Pass by value and pass by refrence

#include <iostream>

using namespace std;

void increment(int a)

{

    a = a+2;

    cout<<"Inside function :"<<a<<endl;

}

int main() {

    int a = 10;

    increment(a);

    cout<<"Inside main :"<<a<<endl;

    return 0;

}

#include <iostream>

using namespace std;

void increment(int \*aptr) // accepting address

{

    \*aptr = \*aptr + 1;

    cout<<"Inside function :"<<\*aptr<<endl;

}

int main() {

    int a = 10;

    increment(&a); // sending

    cout<<"Inside Main :"<<a<<endl;

    return 0;

}

3. Dynamic Memory Allocation

#include <iostream>

using namespace std;

int main() {

    int arr[100]; // static declaration

    // dynamic allocation

    int n;

    cin>>n;

    int \*a = new int[n];

    for(int i=0; i<n; i++)

    {

        cin>>a[i];

        cout<<a[i]<<", ";

    }

    delete []a;

    return 0;

}

4. Insertion at Head Linkedlist

#include <iostream>

using namespace std;

class node{

public:

    int data;

    node \*next;

    node(int d)

    {

        data = d;

        next = NULL;

    }

};

void insertionAtHead(node\*&head, int data)

{

    if(head==NULL)

    {

        head = new node(data);

        return;

    }

    node \*n = new node(data);

    n -> next = head;

    head = n;

}

void print(node\*head)

{

    node \*temp = head;

    while(temp!=NULL)

    {

        cout<<temp->data<<"-->";

        temp = temp->next;

    }

    cout<<endl;

}

int main() {

    node\* head = NULL;

    insertionAtHead(head, 5);

    insertionAtHead(head, 1);

    insertionAtHead(head, 3);

    print(head);

    return 0;

}