Exam no 1

Wed next week

Projects & quizzes refer

1-4

Pointers

Int a=5; a

|  |
| --- |
| 5 |

Int\*p; p=&a

cout<<a; //5

Cout<<\*&a; //5

cout<<\*p; //5

Int b[4]={9,5,8,3};

b[0] b[1] b[2] b[3]

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | 5 | 8 | 3 |

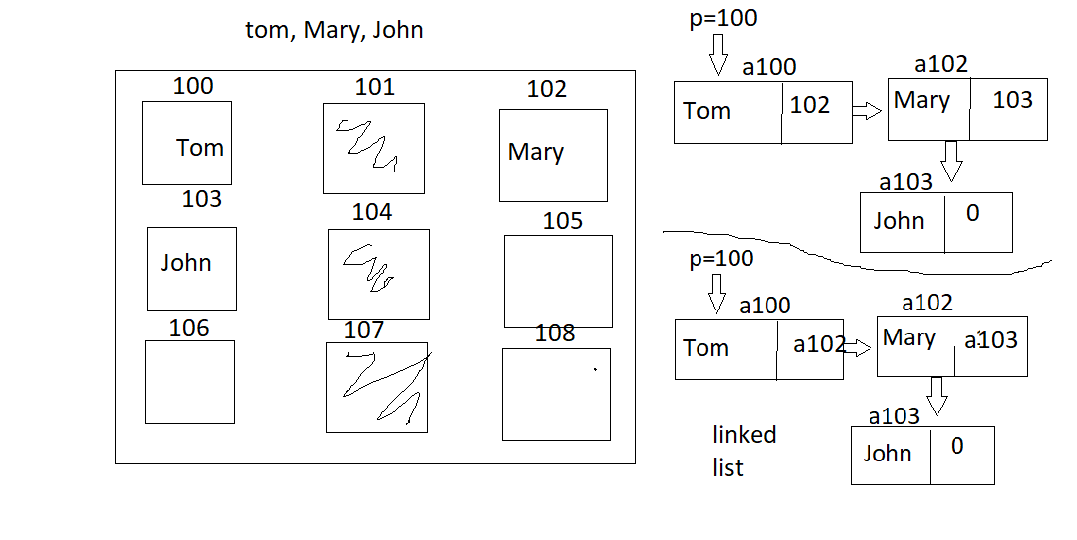
address{ &b[0] &b[1] etc. . .

b+0 b+1 etc. . .

cout<<b[2]; //8

cout<<\*&b[2]; //8

cout<< \*(b+2); //8



to store an array of multiple items we use an array of STRUCTs

struct PERSON

{

string name;

int age;

float gpa;

};

PERSON x[3]; //we have 3 consecutive locations

x[0] x[1] x[2]

|  |  |  |
| --- | --- | --- |
| Tom|20|2.5 |  |  |

//one room for item  
name age gpa

x[0]. //dot operator to get the value of each

x[0].gpa = 2.5

x0].age = 20;

x[0].name=”Tom”;

//with this idea is the same we’ll have to do for Tom Mary and John

//each of these boxes are called a node

|  |
| --- |
| | |

each is called a node

struct node //node I sused for linked list at least that’s standard name

{

string name;

node \*next; //next is a pointer to node :/ another compartment to add the address of next node

};

node \*p; //p is a pointer to node. p can hold an address of a location who is shape and set of //information to the struct node

//we now want to ask the compiler to find the first available nocation

//the thing that will do it is keyword new

new //will help us search the memory to see which location is available. another info new will need is a //location big enough to hold it in

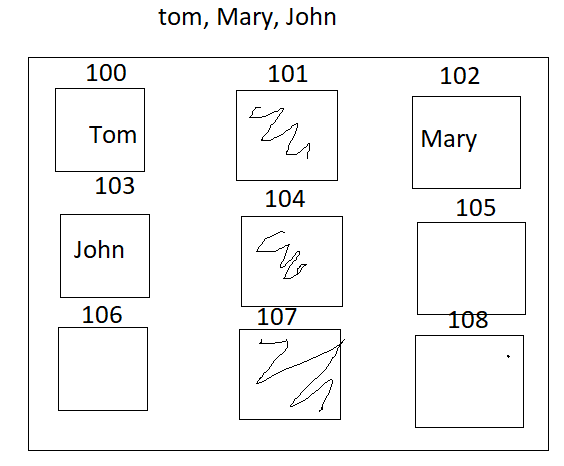
p=new(node); //we want to assign the address of that location to p

ex: a100 //suppose this was the first available location

|  |
| --- |
|  |

🡨 P //p is pointing to it

P.name=”Tom” //but bec P is holding the address of a100

the . will turn into p->name=”Tom”

a100

|  |
| --- |
| Tom | |

p->next=new(node);

a100 a102

|  |
| --- |
| Tom | |

|  |
| --- |
| Mary | |

p->next

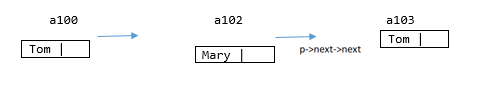
p->next->name=”Mary”;

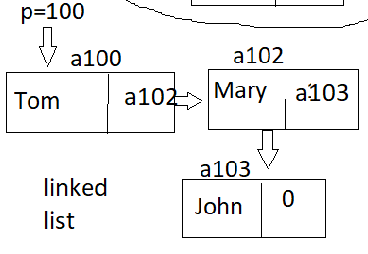
//now room for John

p->next->next=new(node);

a103

|  |
| --- |
| | |





//we always make a copy bec if we lose it

node \*c = p; //we use this key to visit all those nodes

//we don’t use for loop bec we don’t know how many

----------DISPLAYING CONTENT OF ALL NODES----------

while(c!=NULL)

{

cout<<C->name;

c=c->next;

}

//c is pointing to a100 to Tom since name

//then a1000 Tom is pointing to a102 which is Mary, is why c=c->next;

//c is now = a102 and a102 is pointing to a103

//c now becomes c=a103 who is pointing to John, is a103 == 0 or NULL?

//next a103 is pointing to 0, c=0 now pointing to it. c==0 we exit loop

//we do this bec if we want to go back to the beginning we visit p

----------------WRITE TO COUNT THE NUMBER OF NODES------------------

//we need a counter set it to 0 and throw it to visit the nodes, when u visit increase counter by 1

//new is to find you something, to visit the nodes you use that while loop //above

//we don’t want to use p bec we don’t want to lose it

c=p;

int counter=0;

while(c!=NULL)

{

counter++;

//then go to the next node

c=c->next;

}

cout<<counter; //=the number of nodes

//how this thing works. the counter is 0, since c is pointing to a100 and not 0 we increase counter by 1. now c->next c is now pointing to the next which is a102, which is still not 0 so add 1 to counter again. counter == 2 right now, then c=c->next again and we pointing to a103 and add 1 to counter again, then c->next now points to 0 and since c==NULL or 0, we stop the loop and print how many counter was.

//if we have

while(c->next!=NULL)

{

cout<<C->name;

c=c->next;

}

/since c=p, c=a100 then c=a102 print Tom then c->next, is 103==0? no so print Mary, c->next then is 0 then skips over john

//you exit the loop if c is pointing ot the last node

//display the content of the last node

c=p;

while(c->next!=NULL)

{

c=c->next;

}

//is c a100 == to 0? no point ot next is a102, /= 0 still, a103? /=0 still, then points to 0

cout<<c->name; //John

//where is Mary(it. in which node Mary is stored?)

//we prepare to visit all nodes again

c=p;

int nodeNumber=1;

while(c->name!=”Mary”) //we go forward if the content name is not mary

{

c=c->next;

nodeNumber++;

}

//c starts at the first node, is c->a100 name tom equal to mary? no next

c->nextname, is c->name == to mary which is yes

cout<<”Mary is in node#: “<<nodeNumber;

//-------------------project 5----------------------

same with the lecture here

Tom 215 + $

//how will we know that the first one a name then a number

//each time we read that at a time we use a token

stirng token; //read tom as a string, read 215 as a string, + and $ as a //string

while(cin>>token, token!=”$”) //string dollar sign

{

//every time we read something it goes through an array of characters

//and ther eis a \0 at the end

//when we reach the 215 it is a string “215”

//when we read a token we decide whether it is a variable, a number or

//an operation

//the predefined function to change a string of digits to numeric

//use that function to become 215

//tom is =20 for some reason? so we use the postfix expression with the

//switch statements again

}