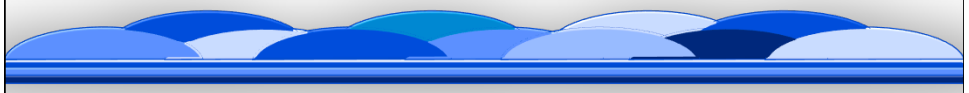
Topic 10b – Searching and Pointers - Review



Searching (brief Review)

- What is the difference between searching from 1 element and searching for the # of instances?
  - ➔ 1 element we search until found
  - ➔ # of instances we search the entire list
- How do we search for 1 element?
  - Sequential search
- How do we search for the # of instances?
  - Search the entire list and count the instances

Searching a list for one instance

```
int *ptr;
bool found;

ptr    = head;
found = false;

while(ptr != NULL &&  !found)
{
       if (ptr->item == searchItem)
       {
           found = true;
       }
       else
       {
            ptr = ptr-> next;
       }
}
```

NOTE: we should make sure we are not at the end of our list AND if we are looking for one element we should stop searching when it is found

3

Searching a list for the # of instances

```
int *ptr;
int instances;

ptr    = head;
instances = 0;

while(ptr != NULL)&&  !found)
{
      if (ptr->item == searchItem)
      {
          instances = instances +1;
      }
      else
      ptr = ptr -> next;
           ptr = ptr-> next;
}
```

What should we change?

We don't need found

We don't need found we need instances

We move to next every time

4

Pointers as a return type

```
PersonNode *CreateList()
{

   return head;
}
```

Write the code to create a link-list of type PersonNode
Assume you are using an input file

5

Pointers can be passed too

```
void OutputList(PersonNode *head)
{
    PersonNode *perPtr;

     perPtr = head;
}
```

Write the code to output a list of type PersonNode

Managing Pointers in Function

- If you need to update a pointer in a function you can
  - Pass is as value and return the pointer

```
PersonNode *StackPush(PersonNode *head,
                      PersonNode newNode)
{
    …
    return head;
}
In main:
    stackTop = StackPush(stackTop, node);
```

7

Managing Pointers in Function

- If you need to update a pointer in a function you can
  - Pass it as reference

```
Void StackPush(PersonNode **head, (or &*head)
               PersonNode newNode)
{
    …
    *head = perPtr;
}
In main:
    StackPush(&stackTop, node);
```

8