Program 1:

```
[fiq8745@omega ~]$ gcc -o ages linked_ages.c
[fiq8745@omega ~]$ ./ages
Enter age 1:
45
Enter age 2:
23
Enter age 3:
90
People less than 21: 0.
```

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node
  {
     int age;
     struct Node *next;
  }Node;
```

(THIS IS JUST AN INTRO-WE WILL LEARN NEXT CLASS HOW TO BUILD A BETTER LINKED LIST)

```
/*see if any value is less than n*/
int check_list(Node*h, int n)
{
  Node*temp;
  Node* current=h; /*start with first node-address held in current*/
  int counter=0;

  while(current) /*keep going until current==NULL*/
  {
          if(current->age<n)
          {
                  counter++;
          }
          temp=current; /*hold the address so we don't lose it (we need to free it)/*
          current=current->next; /*current is now the next node in the list*/
          free(temp);
  }

  return counter;
}

int main(int argc, char **argv)
{
  int n;
```

```
  Node* head  = malloc(sizeof(Node));  /*first node-don't forget to malloc or else head is just a pointer-you will get a seg fault if you try to put information in it*/
  Node* second = malloc(sizeof(Node)); /*second node*/
  Node* third  = malloc(sizeof(Node));  /*third node*/

  if(!head || !second || !third) /*make sure malloc didn't return NULL*/
  {
        printf("Memory not allocated. Exiting...");
  }

  else
  {
        printf("Enter age 1:\n");
        scanf("%d", &head->age); /*put a value in the first node.  Remember head ->age is the same as *head.age (dereferencing head) and then accessing the age member.  the address operator is used in scanf*/
        head->next=second; /*set the next member of our node to the address of the next node-linking up the list*/

        printf("Enter age 2:\n"); /*same as above*/
        scanf("%d", &second->age);
        second->next=third;

        printf("Enter age 3:\n");
        scanf("%d", &third->age);
        third->next=NULL;

        n=check_list(head,21);

        printf("People less than 21: %d\n",n);
  }

}
```