Quiz 4 Version B

Due No due date **Points** 20 **Questions** 12
Available after Nov 11 at 1pm **Time Limit** None **Allowed Attempts** Unlimited

Instructions

Quiz for Lecture 4: Linked Data Structures, Linked Lists, and const

18 points required to pass

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	less than 1 minute	0 out of 20 *

* Some questions not yet graded

Score for this attempt: **0** out of 20 *
Submitted Nov 28 at 2:08pm
This attempt took less than 1 minute.

Question 1

0 / 1 pts

Examine this code. Line numbers in main are specified in comments. In the check boxes below, check each line of code that WILL NOT COMPILE.

```
struct cl_node {   char data;   cl_node *next;   };

int main() {

     const cl_node n1{ 'a', nullptr };       // 2

     cl_node *p1 = &n1;                      // 3
```

```
    const cl_node *p2(&n1);                  // 4

    cl_node *p3 = new cl_node{ 'b', nullptr };      // 5

    const cl_node *p4{p3};                   // 6

    p4->data = 'X';                          // 7

}
```

'ou Answered

☑ Line 2 will NOT compile

orrect Answer

□ Line 3 will NOT compile

□ Line 4 will NOT compile

□ Line 5 will NOT compile

□ Line 6 will NOT compile

orrect Answer

□ Line 7 will NOT compile

Jnanswered

Question 2

0 / 1 pts

What is typically specified as the return type for a function that returns only non-negative integer values?

○ int

○ sizeof

orrect Answer

○ **size_t**

○ **il_node ***

Question 3

Not yet graded / 1 pts

Write the C++ code to define an **dl_node** structure, in which the **data** member is a **double** and the **next** member is a *pointer-to* **dl_node**:

Your Answer:

Jnanswered

Question 4

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

struct cl_node { char data; cl_node *next; };

Which of these would be the most reasonable interface for a function that makes a singly linked list of **char** from a C-string?

○ **cl_node *mk_clist_from_Cstring(cl_node *p) { ... }**

orrect Answer

○ **cl_node *mk_clist_from_Cstring(const char *cstr) { ... }**

○ **void mk_clist_from_Cstring(cl_node *p) { ... }**

○ **cl_node *mk_clist_from_Cstring(const char *cstr, cl_node **pp) { ... }**

○ cl_node **mk_clist_from_Cstring(const char *cstr) { ... }

○ void mk_clist_from_Cstring(const char *cstr) { ... }

Jnanswered

Question 5

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

struct cl_node {   char data;   cl_node *next;   };

Which of these would be the most reasonable interface for a function that modifies the data in one or more of the linked list nodes, but does not modify the shape of the linked list?

○ void clist_mod_data(cl_node **pp) { ... }

orrect Answer

○ **void clist_mod_data(cl_node *p) { ... }**

○ void clist_mod_data(const cl_node *p) { ... }

Jnanswered

Question 6

0 / 1 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node {   int data;   il_node *next;   };

Which of these would be the most reasonable interface for a function that accesses the data in one or more of the linked list nodes, but does not modify any data and does not modify the shape of the list?

○ **int ilist_access_data(il_node **pp) { ... }**

○ **int ilist_access_data(il_node *p) { ... }**

orrect Answer

○ **int ilist_access_data(const il_node *p) { ... }**

Question 7

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node { int data; il_node *next; };

Write the body of code for this function, that returns the number of nodes in the list of integers that are less than or equal to **ceiling**:

```
size_t ilist_size(const il_node *p, int ceiling)

{

    // this is the part you have to write as your answer

}
```

Your Answer:

```
size_t count(0);

for ( ; p; p = p->next)

   if (p->data <= ceiling)

      count += 1;

return count;
```

Question 8

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

struct il_node {   int data;   il_node *next;   };

Write the body of code for this function, that returns **true** if all the data values in the list are <= 0, and **false** otherwise:

```
bool ilist_all_le_0(const il_node *p)

{

      // this is the part you have to write as your answer

}
```

Your Answer:

```
for ( ; p; p = p->next)
    if (p->data > 0)
        return false;
return true;
```

Question 9

Not yet graded / 3 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

```
struct cl_node {   char data;   cl_node *next;   };
```

Write the body of code for this function, that makes a duplicate (separate) copy of the list of **char**s pointed to by **p**, and returns the address of the first **cl_node** in the copy.

```
cl_node *mk_clist_copy(const cl_node *p)
{
        // this is the part you have to write as your answer
}
```

Your Answer:

```
cl_node *pcopy(0);

cl_node **ppcopy(&pcopy);

for ( ; p; p = p->next) {

   *ppcopy = new cl_node{ p->data, nullptr };

   ppcopy = &(*ppcopy)->next;

}

return pcopy;
```

Question 10

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **double**:

struct dl_node {   double data;   dl_node *next;   };

Write the body of code for this function, that appends a new **dl_node** with **data** set to **value** at the end of the list of **double**.

```
void dlist_append_value(dl_node **pp, double value)

{

      // this is the part you have to write as your answer

}
```

Your Answer:

```
while (*pp)
    pp = &(*pp)->next;
*pp = new dl_node{ value, nullptr };
```

Question 11

Not yet graded / 3 pts

Suppose we have this structure definition, for a node in a singly linked list of **int**:

```
struct il_node {   int data;   il_node *next;   };
```

Write the body of code for this function, that returns a pointer to the first **il_node** in a list of **int** values, where the list consists of **n** nodes, each with data set to **val**.

```
il_node *mk_ilist_of_n_val(int val, size_t n)
{
    // this is the part you have to write as your answer
}
```

Your Answer:

```
il_node *head = nullptr;

il_node **phead = &head;

for ( ; n; --n) {

   *phead = new il_node{ val, *phead };  // insert is easier than append

return head;
```

Question 12

Not yet graded / 2 pts

Suppose we have this structure definition, for a node in a singly linked list of **double**:

```
struct dl_node {   double data;   dl_node *next;   };
```

Write the body of code for this function, that returns the maximum value stored in the list of **double**s. (Return **0.0** if the list is empty.)

```
double dlist_max(dl_node *p)

{

       // this is the part you have to write as your answer

}
```

Your Answer:

```
if (p == nullptr)
   return 0.0
double max = p->data;
for (p = p->next ; p; p = p->next)
   if (p->data > max)
      max = p->data;
return max;
```

Quiz Score: **0** out of 20