.

COMP 2012H Midterm Exam - Fall 2018 - HKUST

Date: October 20, 2018 (Saturday)

Time Allowed: 2 hours, 10am–12nn

Instructions:

1. This is a closed-book, closed-notes examination.
2. There are **7** questions on **28** pages (including this cover page and 4 blank pages at the end).
3. Write your answers in the space provided.
4. All programming codes in your answers must be written in the ANSI C++ version as taught in the class.
5. For programming questions, unless otherwise stated, you are **NOT** allowed to define additional structures, classes, helper functions and use global variables, **auto**, nor any library functions not mentioned in the questions.
6. Approved calculators are allowed for this exam.

Student Name	
Student ID	
Email Address	
Venue & Seat Number	

For T.A. Use Only

Problem	Topic	**Score**
1	True or False	/ 10
2	Function Parameter Passing Mechanism I	/ 7
3	Function Parameter Passing Mechanism II	/ 5
4	Structure and Object	/ 8
5	Recursion	/ 11
6	Circular Doubly Linked List	/ 24
7	Pointer and Dynamic Array	/ 35
Total		/ 100

Problem 1 [10 points] True or false

Indicate whether the following statements are *true* or *false* by circling **T** or **F**. You get 1.0 point for each correct answer, −0.5 for each wrong answer, and 0.0 if you do not answer.

T **F** (a) The following program:

```
#include <iostream>
using namespace std;

int main() {
  int a = 10, b = 20, c = 30;
  if(a < 10)
    if(b > 15)
      cout << "Point A" << endl;
  else if(c < 30)
    cout << "Point B" << endl;
  else
    cout << "Point C" << endl;
}
```

can compile with NO errors and run, and it prints `Point C`.

T **F** (b) Given the following program:

```
#include <iostream>
using namespace std;

enum classification { FIRST, SECOND_UPPER = -2, SECOND_LOWER, THIRD, PASS };

int main() {
  cout << "FIRST: " << FIRST << endl;
  cout << "SECOND_UPPER: " << SECOND_UPPER << endl;
  cout << "SECOND_LOWER: " << SECOND_LOWER << endl;
  cout << "THIRD: " << THIRD << endl;
  cout << "PASS: " << PASS << endl;
}
```

The output of the program is

```
FIRST: -1

SECOND_UPPER: -2

SECOND_LOWER: -1

THIRD: 0

PASS: 1
```

T **F** (c) The following program CANNOT be compiled successfully.

```
#include <iostream>
using namespace std;

void print(int num);

int main() { print(); }

void print(int num = 10) {
  for(int i=0; i<num; i++)
    cout << "Printing..." << endl;
}
```

T **F** (d) The following code can be compiled WITHOUT errors.

```
int arr1[] = { 1, 2, 3, 4, 5 };
int arr2[] = { 6, 7, 8, 9, 10 };
arr1 = arr2;
```

T **F** (e) Assume an array is declared as follows:

```
int arr[] = { 1, 2, 3 };
```

The following statements all print the same value on screen.

```
cout << arr << endl;
cout << &arr << endl;
cout << &arr[0] << endl;
```

T **F** (f) The following program prints the address of `str` on screen.

```
#include <iostream>
using namespace std;
int main() {
  char str[] = "Peter";
  cout << str << endl;
}
```

T **F** (g) The following program CANNOT be compiled successfully.

```
int main() {
  int* const p = new int;
  const int* q = p;
  delete p;
}
```

T **F** (h) The following program will result in memory leak.

```
int main() {
  int* p;
  for(int i=0; i<10; i++) {
    p = new int[10];
  }
  delete [] p;
}
```

T **F** (i) The following program can be compiled WITHOUT errors.

```
#include <iostream>
#include <string>
using namespace std;

struct Person {
  string name;
  char gender;
  int age;
  int marks[5];
};

int main() {
  Person bingyen, guangneng, jingyang;
  bingyen = guangneng = jingyang;
}
```

T **F** (j) The following program can be compiled WITHOUT errors.

```
struct MyArray {
  const int size;
  int* arr;
};

int main() {
  MyArray myArr { 10, new int[10] };
  delete [] myArr.arr;
}
```

Problem 2 [7 points] Function Parameter Passing Mechanism I

```
1  #include <iostream>
2  using namespace std;
3
4  int mystery(int x, int& y, int z) {
5    x = z += 5;
6    y = x + y + z;
7    return y;
8  }
9
10 int main() {
11   int a = 1, b = 2, c = 3;
12
13   /* This comment line is to be replaced by the statement in the question.  */
14
15   return 0;
16 }
```

If line #13 of the above program is replaced by each of the following statements, give the output if the resulting program can be compiled; otherwise, explain the compilation error.

(a) `cout << mystery(a, b, c) << endl;`

Answer:____

(b) `cout << mystery(mystery(b, b, b), b, c) << endl;`

Answer:____

(c) `cout << mystery(a, mystery(a, b, c), c) << endl;`

Answer:____

Problem 3 [5 points] Function Parameter Passing Mechanism II

C++ provides plus equal operator, +=, for integers. Implement this operator by a function `plus_equal` so that the following program

```
#include <iostream>
using namespace std;

/* The definition of function plus_equal will be put here */

int main() {
  int a = 10;
  int b = 20;

  plus_equal(a, plus_equal(b, 5)); // Equivalent to a += b += 5;

  cout << "a: " << a << ", b: " <<  b << endl;
}
```

will give the output below:

```
a: 35, b: 25
```

Remark:

i. You are not allowed to use C++'s built in (plus equal) += operator in your answer.

ii. You have to decide the exact function header of the `plus_equal` function yourselves.

Answer:

Problem 4 [8 points] Structure and Object

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  struct EnglishWord {
6    char* str = nullptr; // You are NOT allowed to modify this line
7  };
8
9  void print(EnglishWord& ew);
10
11 // Initialize the EnglishWord ew with the given C string s
12 void init(EnglishWord& ew, char* s) {
13   ew.str = s;
14   print(ew);
15 }
16
17 // You are NOT allowed to modify this function
18 void remove(EnglishWord& ew) {
19   print(ew);
20   delete [] ew.str;
21 }
22
23 void print(EnglishWord& ew) {
24   cout << ew.str << endl;
25 }
26
27 // You are NOT allowed to modify the main function
28 int main() {
29   char word[] = "Tricky";
30   EnglishWord w1, w2, w3;
31   init(w3, word);
32   w1 = w3;
33   remove(w3);
34   remove(w2);
35   remove(w1);
36 }
```

The program above will compile but it runs with 3 run-time errors during the remove function calls for the 3 `EnglishWord` objects, `w3`, `w2` and `w1` in that order.

Some remarks:

- *You are NOT allowed to change the meaning of any given global function.*
- *You are NOT allowed to modify the main function nor the remove function in your answers to this question.*
- *You don't get ANY marks if you only give the line numbers without correction for parts (a) and (b), or explanation for part (c).*

(a) [3 points] Identify the statement, by giving its line number in the program, that causes the run-time error when `w3` is destructed. Re-write the concerned statement to eliminate this error.

Answer:

(b) [3 points] However, even after you fix the error in part (a), it still will run into another run-time error when `w2` is destructed. Again, identify the statement, by giving its line number, that causes the error, and then re-write the concerned statement to eliminate this 2nd error.

Answer:

(c) [2 points] To your dismay, even after you fix the 2 errors in part (a) and (b), your program still will run into the 3rd run-time error when `w1` is destructed. Identify, for the last time, the statement, by giving its line number, that causes this last error. This time, you only need to explain how the error is produced and you don't need to fix it.

Answer:

Problem 5 [11 points] Recursion

Answer:

```
/* - int data[10][10]: A 10x10 2D array with certain number of 1s.
   - int x, int y: Starting point (x, y).
   - int v: The value to be processed in the current round.
   - int mark[10][10]: A 10x10 2D array that records the processed locations
                       and values so far.

   Note: You can assume the 2D array, data, is always valid, only contains
   1s and 0s, and there will only be one distinct group of connected 1s.
*/

void spreadout(int data[10][10], int x, int y, int v, int mark[10][10]) {
  if(x < 0 || x >= 10) return;
  if(y < 0 || y >= 10) return;

  // WRITE YOUR CODE HERE




}
```

Problem 6 [24 points] Circular Doubly Linked List

(a) [4 points] Implement `CDLL create(int value)`.

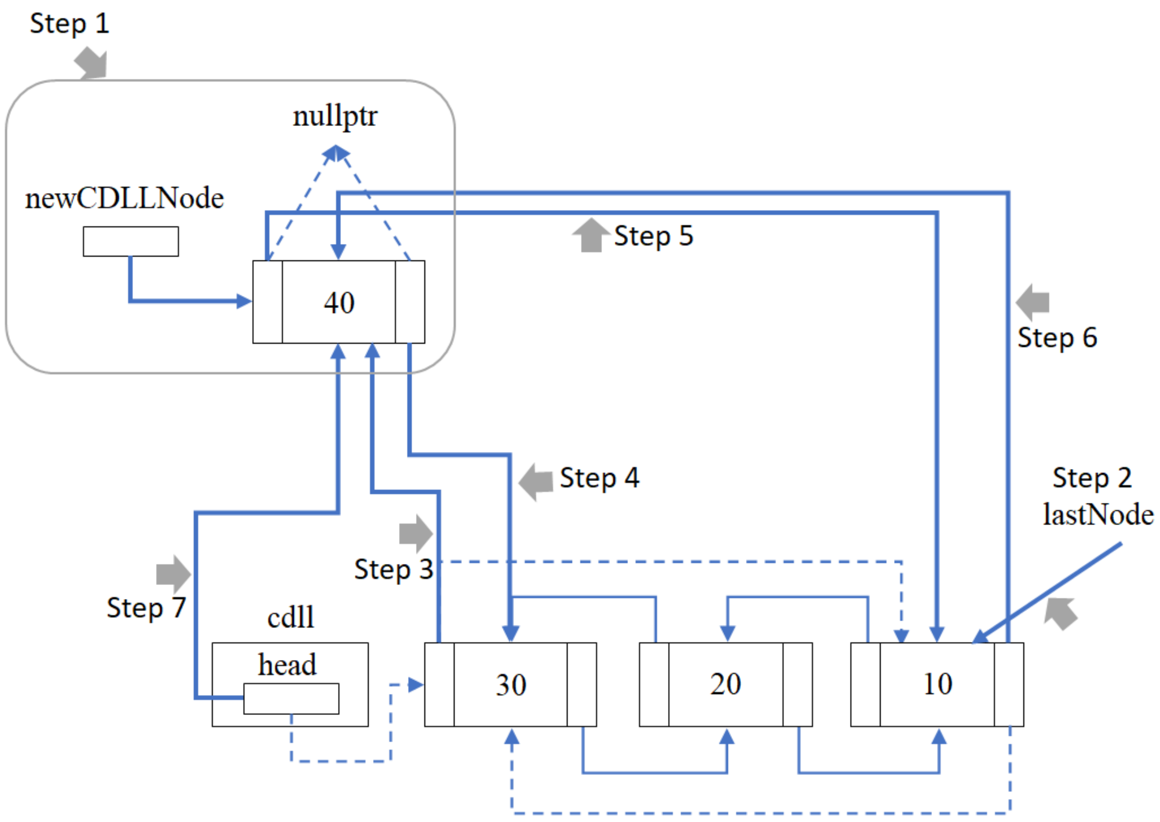
Answer:

(b) [1 point] Implement `bool isEmpty(const CDLL& cdll)`.

Answer:

(c) [7 points] Implement `void insertAtFront(CDLL& cdll, int value)`.

Hint: You may want to implement the function according to the example given below.



Answer:

(d) [12 points] Implement `int removeFromBack(CDLL& cdll)`.

Remark: Make sure to handle all the special cases.

Answer:

Problem 7 [35 points] Pointer and Dynamic Array

(a) [3 points] Implement the `Dictionary` structure.

Answer:

(b) [2 points] Implement `void init(Dictionary& dict)`.

Answer:

(c) [3 points] Implement `int findWord(const Dictionary& dict, const string& word)`.

Hint: Two string objects can be compared using equality operator (i.e. ==). It returns true if the two strings are the same. Otherwise, it returns false.

Answer:

(d) [12 points] Implement `bool insertWord(Dictionary& dict, const string& word)`.

Note: You can use the relational operators, <, <=, ==, >=, > for lexicographic string comparison.

Answer:

(e) [10 points] Implement `bool removeWord(Dictionary& dict, const string& word)`.

Answer:

(f) [2 points] Implement `void displayDict(const Dictionary& dict)`.

Answer:

(g) [3 points] Implement `void destroy(Dictionary& dict)`.

Answer:

-------------------- END OF PAPER --------------------