MD2
(29)

THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF JULY 2018

Code and Name of Course: COMP1603 — Computer Programming III Paper:

Date and Time: Wednesday 18th July 2018 9 am Duration: 2 Hours

INSTRUCTIONS TO CANDIDATES: This paper has 6 pages and 3 questions

Answer all questions

1. Examine the following program. Assume that it compiles without error.

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

struct Stock{
    string name;
    double perShare;
};

void setshareValue(Stock *p, double perShare){
    p->perShare = perShare;
}

int main(){
    Stock a, b;
    Stock *ap, *bp;

    double perShare = 11.67;
    double *newShare =(double *) malloc(sizeof (double));
    *newShare = 2.00 + *(&perShare);
    cout<<"1. New share value is $ "<<*newShare<<endl;

    ap=&a;
    a.name = "Apple";
    a.perShare = 187.83;
    ap->perShare = ap->perShare - 0.53;
    cout<<"2. "<<a.name<<" share value is $"<<a.perShare<<endl;

    b.name = "Google";
    b.perShare = 1162.81;
    bp = (Stock *) malloc (sizeof (Stock));
    *bp = b;
    b. perShare = 1162.49;
    cout<<"3. "<<(*bp).name<<" share value is $"<<bp->perShare<<endl;

    a.name = "Apple";
    a.perShare = 188.87;
    setshareValue(ap, 188.02);
    cout<<"4. "<<a.name<<" share value is $"<<a.perShare<<endl;

    ap->name = "Apple Inc.";
    ap->perShare =190.00;

    bp = ap;
    setshareValue(bp,1165);
    cout<<"5. "<<bp->name<<" share value is $"<<bp->perShare<<endl;
}
```

Question 1 continues on the next page

a. The `main()` function contains the following declarations:

```
Stock a, b;
Stock *ap, *bp;
```

Explain clearly the difference between the variables `a` and `ap`. [2 marks]

b. Write down the output generated by the first five (5) `cout` statements. [8 marks]

c. In the main function, `a` and `b` are two variables of type `Stock`. Write a function, `swap`, which given the two variables `a` and `b`, interchanges the values of `a` and `b` so that the change is known in `main`. [4 marks]

d. Show how you would call the `swap` function in `main`. [1 mark]

Total Marks: 15

2. a. What is an Abstract Data Type (ADT)? [2 marks]

b. Convert the following infix expressions to postfix expressions:

i. $a * (b - c + d) - e / f$

ii. $(7 - 8 / 2 / 2) * ((7 - 2) * 6)$ [4 marks]

c. Assume that the following functions for stack operations are available:

```
Stack * initStack();
bool isEmpty (Stack * s);
bool isFull (Stack * s );
void push (Stack * s, char ch);
char pop (Stack * s);
```

You do *not* need to *write the code* for these functions.

Question 2 continues on the next page

Use the above functions, to write code to prompt the user for a string, and determine whether the string contains balanced parentheses, that is, whether each left parenthesis (if there is any) has exactly one matching right parenthesis appearing later in the string. Your code must print a line indicating the number of parentheses which were in the expression, followed by one of the following specific error messages:

- *Balanced parentheses*
- *More left than right parentheses*
- *More right than left parentheses*

You may assume that the string is stored as an array of characters ending with '\0'.

[9 marks]

Total Marks: 15

3. a. A linked list is required to store a set of characters of a word, where each node of a linked list stores one character.

The following is the declaration of a Node in the linked list:

```
struct Node {
    char ch;
    Node *next;
};
```

Two words are *anagrams* if one word can be formed by rearranging all the letters of the other word for example sister and resist. A word is represented as a linked list as defined above.

Write a function which given `top1` and `top2` each pointing to a word of lowercase letters returns *true* if the two words are anagrams and *false* if they are not.

Question 3 a. continues on the next page

Your function must be based on the following:

For each letter in `top1`, search `top2` for it; if the letter is found delete the node from `top2` and continue; otherwise return *false*. If `top2` is empty after the traversal of `top1` is completed then return *true*. [9 marks]

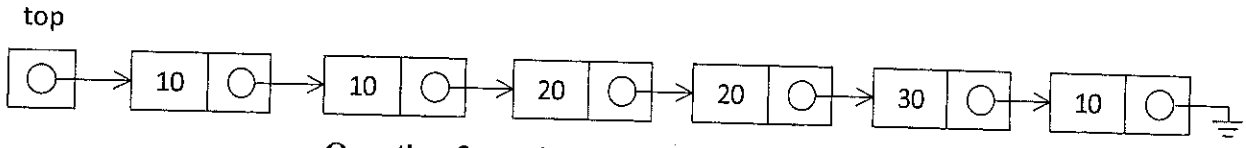
b. Write a recursive function, `int occurrences (Node * top, char ch)` which takes the address of the first node of a linked list of characters and a character `ch` as parameters and returns the number of occurrences of `ch` in the linked list. [5 marks]

c. Consider the following function, `mystery`:

```
Node* mystery(Node * top, int n){
   if(top ==NULL)
      return NULL;
   Node *rest = top -> next;
   if(top->data == n){
      top->next = mystery(rest,n);
      return top;
   }
   else
      return mystery(top->next, n);
}
```

Given the linked list shown below, `mystery` is called with top and the value 10.

Draw the linked list returned by the call `mystery (top, 10)`. [6 marks]



Question 3 continues on the next page

d. What is the output produced by the call `w(1)` of the following recursive function?

Show your working.

```
void w (int n){
    cout<< n <<" ";
    if (n < 9){
        w(n + 2);
        cout<<n<<" ";
    }// end if
}
```

[5 marks]

Total Marks: 25

Total Marks: 55

END OF EXAMINATION