UCLA Computer Science Department TA: Kung-Hua Chang

Student Name and ID _____

CS 31, SPRING 2014, PRACTICE FINAL EXAM.

Problem	Maximal Possible Points	Received
1	60	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	15	
9	15	
Total Score	150	

Problem #1: There is a Vending Machine Class called VM below. Please complete the codes under the comment TODO. The output of this program is:
I bought Coke
Pepsi is sold out!!

```
#include <iostream>
#include <string>
using namespace std;
const int MAXSODA = 100;

class Soda
{
public:
        Soda();
        void setName(string name);
        string getName() const;
private:
        string name;
};

Soda::Soda()
{
        // TODO: set name = "NA" to signal Not Available (5 points)
        this->name = "NA";
}
void Soda::setName(string name)
{
        // TODO: set Soda's name as the name passed in. (5 points)
        this->name = name;
        // We have to use this->name here.
}
string Soda::getName() const
{
        // TODO: return Soda's name. (5 points)
        return name;
        // or return this->name;
}
```

```
class VM
{
public:
	VM(int n);
	~VM();
	void restock(string name,int quantity);
	Soda* getSoda(string name);
	bool buySoda(string name);
private:
	Soda* inventory[MAXSODA];
	int quantity[MAXSODA];
	int numSoda;
};

VM::VM(int n)
{
	numSoda = n;

	for(int i=0;i<numSoda;i++)
		inventory[ i ] = new Soda();
}

VM::~VM()
{
	// TODO: (10 points)
	// Delete the storage pointed to by Soda pointers in inventory array

	for(int i=0;i<numSoda;i++)
		delete inventory[i];

}
```

```
void VM::restock(string name,int quantity)
{
        for(int i=0;i<numSoda;i++)
                if( inventory[i]->getName() == "NA") {
                        // TODO: (10 points)
                        // 1. If we found a Soda that has the name "NA", we set
                        // this Soda to have the name we passed into this function.
                        // 2. Set the quantity for that Soda.
                        // 3. Break out of this loop.
                        inventory[i]->setName(name);

                        this->quantity[i] = quantity;

                        break;   // or return;
                }
}

Soda* VM::getSoda(string name)
{

        // TODO: (10 points)
        // Search all Soda Objects to see if there's a matching Soda by name
        //     If there is a matching soda by name, return that Soda object.
        //     Return nullptr if there's no matching Soda by name.

        for(int i=0;i<numSoda;i++)
                if( inventory[i]->getName() == name)
                        return inventory[i];

        return nullptr;


}
```

```
bool VM::buySoda(string name)
{
        // TODO: (15 points)
        // 1. Search through all Soda objects to find matching Soda by name
        //     If there is a matching soda by name, and if the quantity is > 0.
        //     then we decrease quantity for the Soda by 1 and return true
        //     If there is a matching soda by name, but the quantity is <= 0
        //     then we return false.
        //     If there is no matching soda name, return false.

        for(int i=0;i<numSoda;i++)
                if( inventory[i]->getName() == name && quantity[i] > 0)
                {
                        quantity[i]--;
                        return true;
                }

                return false;
}
int main()
{
        VM vm(5);
        vm.restock("Coke",4);
        vm.restock("Diet Coke",5);
        vm.restock("Sprite",1);
        vm.restock("Pepsi",0);
        vm.restock("Lemonade",1);

        if(vm.buySoda("Coke"))
        cout << "I bought " << vm.getSoda("Coke")->getName() << endl;
        else
        cout << "Coke is sold out!!" << endl;

        if(vm.buySoda("Pepsi"))
        cout << "I bought " << vm.getSoda("Pepsi")->getName() << endl;
        else
        cout << "Pepsi is sold out!!" << endl;

        return 0;
}
```

Problem #2: Given two strings str1 and str2, str1 is the permutation of str2 if all the characters in str1 appear in str2 but in different order. For example, “12345” is the permutation of “54132”. Assume that only ‘1’ – ‘9’ will appear in the string. A student coded the following program to solve this problem, but there are something wrong. Please find all the bugs in this program. (C)

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

bool isPermutation(string str1, string str2)
{
        if(str1.size() != str2.size())
                return false;

        int i, j, counts[10];

        for(i=0;i<10;i++)
                counts[i] = 0;

        for(i=0;i<str1.size();i++)
                counts[ str1[i] -   0  ] ++; // (1) Should be counts[str1[i]- '0']++;

        for(i=0;i<str1.size();i++)
                counts[ str2[i] -   0  ] --;   // (2) Should be counts[str2[i]- '0']--;

        for(i=0;i<10;i++)                    // (3)
                if(counts[i] != 0)           // (4)
                        return true;         // (5) Should be return false;
        return false;                        // (6) Should be return true;
}
```

```
int main()
{
        cout << isPermutation("12345","54321") << endl;
        cout << isPermutation("12345","98765") << endl;
        return 0;
}
```

The bugs are in?

(A) 123456
(B) 12356
(C) 1256
(D) 234
(E) 2356
(F) 256
(G) 356
(H) 3
(I) 12
(J) 34

Problem #3: A student implemented a function called deleteB(char *msg) to delete all the B in a C-string. Can you please help evaluate his codes below and tell him what is the output of this program? (2)

```
#include <iostream>
using namespace std;
void deleteB(char *msg)
{
        char *ptr;
        while(*msg != 0)
        {
                if(*msg =='B' || *msg =='b')
                {
                        ptr = msg;

                        while (*ptr != 0)
                        {
                                *ptr=*( ptr + 1 );
                                ptr++;
                        }
                }
                msg++;
        }
}

int main()
{
        char msg[100] = "BaabbaBaB.";
        deleteB(msg);
        cout << msg;
}
```

(1) aaaa.
(2) aabaa.
(3) BaaaaB.
(4) BabaBaB.
(5) aabbaa.
(6) aaaaB.
(7) Baaaa.
(8) aaaBa.

Problem #4: What is the output of the program below? (K)

```
#include <iostream>
#include <string>
using namespace std;
int foo( string s )
{
        if (s.size () < 1)
                return -1;

        int mult = 1;
        int offs = 0;
        switch (s[0])
        {
                case '-':
                        mult = -1;
                        break ;
                case '+':
                        mult = 1;
                        break ;
                default :
                        offs = 1;
                        break ;
        }
        return mult * (s[1 - offs ] - '0');
}

int foo2(string exp)
{
        string exp1="", exp2="", exp3="";
        int i;

        for(i=0;i<=1;i++) exp1 += exp[i];
        for(i=5;i<=7;i++) exp2 += exp[i];
        for(i=8;i<=13;i++) exp3 += exp[i];

        return foo(exp1) + foo(exp2) + foo(exp3);
}
```

```
int main()
{
        cout << foo2("-3+3+555+99999") << endl;
        return 0;
}
```

(A)1
(B)2
(C)3
(D)4
(E) 5
(F) 6
(G)7
(H)8
(I) 9
(J) 10
(K)11
(L) 12

Problem #5: Below is a zurt class definition. In the main function, we declare a const Zurt * pointer pointing to a zurt object. At the <BLANK> below, which member functions can we call without causing compilation error? For example, can we use zp->health(); ? There are more than one answers to this question. Please find all the answers. (C)(D)(E)

```
#include <iostream>
using namespace std;
class Zurt
{
public:
        Zurt() { m_health = m_row = m_col = 0; }
        Zurt(int health, int r,int c) {
                m_health = health;
                m_row = r;  m_col = c;
        }
        int health() const { return m_health; }
        int row() const { return m_row; }
        int col() const { return m_col; }
        void setHealth(int health) { m_health = health; }
        void setRow(int r) { m_row = r; }
        void setCol(int c) { m_col = c; }
private:
        int m_health, m_row, m_col;
};

int main()
{
        Zurt z(100,1,1);
        const Zurt* zp = &z;

        <BLANK>

        return 0;
}
```

(A) zp->Zurt(); (B) zp->Zurt(100,1,2); (C) int h = zp->health();
(D) int r = zp->row(); (E) int c = zp->col(); (F) zp->setHealth(50);
(G) zp-> setRow(2); (H) zp->setCol(2); (I) zp->m_col = 5;

Problem #6: Below is a zurt class definition. In the main function, we declare a Zurt * pointer pointing to a zurt object. At the <BLANK> below, which member functions can we call without causing compilation error? For example, can we use zp->health(); ? There are more than one answers to this question. Please find all the answers. (C)(D)(E)(F)(G)(H)

```
#include <iostream>
using namespace std;
class Zurt
{
public:
        Zurt() { m_health = m_row = m_col = 0; }
        Zurt(int health, int r,int c) {
                m_health = health;
                m_row = r;  m_col = c;
        }
        int health() const { return m_health; }
        int row() const { return m_row; }
        int col() const { return m_col; }
        void setHealth(int health) { m_health = health; }
        void setRow(int r) { m_row = r; }
        void setCol(int c) { m_col = c; }
private:
        int m_health, m_row, m_col;
};

int main()
{
        Zurt z(100,1,1);
        Zurt* zp = &z;

        <BLANK>

        return 0;
}
```

(A) zp->Zurt(); (B) zp->Zurt(100,1,2); (C) int h = zp->health();
(D) int r = zp->row(); (E) int c = zp->col(); (F) zp->setHealth(50);
(G) zp-> setRow(2); (H) zp->setCol(2); (I) zp->m_col = 5;

Problem #7: What is the output of the program below? 5

```
#include <iostream>
using namespace std;
int main()
{
	int arr[12] = {1,3,5,0,7,2,0,4,4,0,8,8};
	int count = 0;
	for(int i=0;i<11;i++) {
		if(arr[i] = arr[i+1] )
			count++;
		else
			count--;
	}
	cout << count << endl;
}
```

Problem #8: What is the output of the program below? 21

```
#include <iostream>
using namespace std;
int main()
{
	int arr[100] = {1,1,2,3,5,8};
	int *p = (arr+6);
	for(int i=0;i<2;i++) {
		*p = *(p-1) + *(p-2);
		p++;
	}
	cout << *(p-1) << endl;
}
```

Problem #9: What is the output of the program below? 2 3 1 4 8 6 5

```
#include <iostream>
using namespace std;
void swap2(int* a, int *b)
{
   int temp = *a;   *a = *b;   *b = temp;
}
int main()
{
	int arr[7] = { 2,4,6,8,1,3,5 };
	int* first = arr;
	int* last = arr+6;
	int divider = 4;

	while(first < last) {
		while( (first < last) && (*first < divider) )  first++;
		while( (first < last) && (*last  > divider) )  last--;
		swap2(first,last);
	}
	for(first = arr ; first < arr+7; first++)
		cout << *first << " ";
	cout << endl;
}
```