TIC1001—Introduction to Computing and Programming
National University of Singapore

Final Exam - AY18/19

Question 1: C/C++ Expressions

There are several parts to this question which are to be answered independently and separately. Each part consists of a fragment of C/C++ code. Write the exact output produced by the code in the answer box. If an error occurs, or it enters an infinite loop, state and explain why. You may show workings outside the answer box in the space beside the code. Partial marks may be awarded for workings if the final answer is wrong. Assume that all appropriate preprocessor directives e.g., #include < iostream > , etc. have already been defined.

In [ ]:
```
#include <iostream>
using namespace std;
```

A.

In [ ]:
```
string s = "hippo";

string t = s;

for (int i = 0; i < 5; i++)
    t[i] = s[4-i];

cout << (s + t) << endl;
```

i = 0,
t[0] = s[4-0]
s: hipp**o** , t: **o**ippo

i = 1,
t[1] = s[4-1]
s: hip**p**o , t: o**p**ppo

i = 2,
t[2] = s[4-2]
s: hi**p**po , t: op**p**po

i = 3,
t[3] = s[4-3]
s: h**i**ppo , t: opp**i**o

i = 4,
t[4] = s[4-4]
s: **h**ippo , t: oppi**h**

Output: "hippooppih"

B.

In [ ]:
```
int i = 10;

while (true)
{
    cout << i << " ";
    if (i > 5) i -= 2;
    if (i % 3)
    {
        i += 1;
        continue;
    }
    else break;
}
```

i = 10, **cout: 10**
i > 5, i -= 2 = 8
i % 3 = 2, i += 1 = 9

i = 9, **cout: 9**
i > 5, i -= 2 = 7
i % 3 = 1, i += 1 = 8

i = 8, **cout: 8**
i > 5, i -=2 = 6
i % 3 = 0
Break, exit from loop.

Output: 10 9 8

C.

In [ ]:
```
double f(int x)
{
    return x/2;
}

int main()
{
    double d = 15;
    cout << f(d/2)/2;
}
```

Initialize
d = 15

f(15/2)/2
Since d is double, so f(7.5)/2

However `double f(int x)` takes in integer and outputs double, so
Compute 7/2 in integer = 3, then return as double 3.00

Cout 3.00/2, which is 1.5

Ans: 1.5

D.

In [ ]:
```
int &swap(int &a, int &b)
{
    a = b;
    b = a;
    return a;
}

int main()
{
    int x = 1, y = 2;
    int &z = swap(x, y);
    z = 3;
    printf("%d,%d,%d", x, y, z);
}
```

Initialize
x = 1, y = 2

`swap(&x,&y)` ,
x = 2, y = 2.
Return **a** which is the reference to x, which is &x.
&z is now &x, i.e. z and x now shares the *same memory address*.

Update z to 3, x will also be updated to 3.

Output: 3,2,3

E.

In [ ]:
```
int foo(int n)
{
    switch (n)
    {
        case 0:
            n += 1;
        case 2:
            return n/2;
        case 1:
            n += 3;
            break;
        default:
            n -= 1;
    }
    return n*2;
}

int main()
{
    cout << foo(0) << endl;
    cout << foo(2) << endl;
    cout << foo(4) << endl;
}
```

foo(0)
Case 0,
n += 1 = 1, continue
Case 2,
return 1/2 = 0
cout: 0

foo(2)
Case 2,
return 2/2 = 1
cout: 1

foo(4)
Default Case,
n -= 1 = 3
return 3 * 2 = 6
cout: 6

Output:
0
1
6

Question 2: Mystery Boxes

In a game of Mystery Boxes, contestants are given a sequence of boxes, each containing either a positive or negative cash amount. For every box that the contestant opens, the amount of cash will be added to the contestant's current total (a negative amount will result in the total decreasing). Contestants can decide at any time to stop opening boxes and walk away with the current total as their winnings. Contestants start with $0 and the game ends if the current total reaches below \$0 and the contestant will be forced to leave the game with no winnings. The function `int open_box()` simulates the contestant opening a box and returns the cash amount contained in the box. You may assume it is provided.

A.

Suppose there are unlimited number of boxes to open and the contestant simply opens boxes until his winnings falls below zero and he is forced to end. Implement a function `int num_boxes()` that takes no inputs and returns the total number of boxes that the contestant had opened.

In [ ]:
```
int num_boxes()
{
    int total = 0, count = 0;
    while( total >= 0)
    {
        total += open_box();
        count ++;
    }
    return count;
}
```

B.

Hindsight is 20/20. Implement a function `int max_winnings()` that takes no input, and returns the theoretical maximum winnings the contestant would have won had he chosen to stop at the correct time.

In [ ]:
```
int max_winnings()
{
    int total = 0, max = 0;
    while (total >= 0)
    {
        total += open_box();
        if (max > total) max = total;
    }
    return max;
}
```

C.

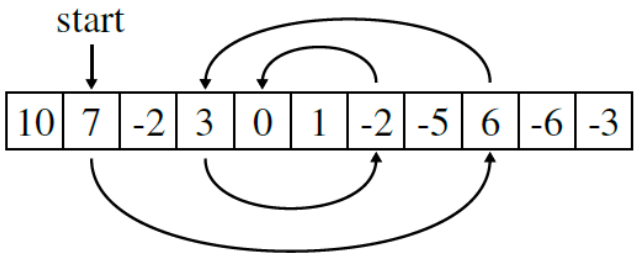
Suppose a contestant wants to open n boxes. He might not be able to do so if the game ends too early. If the rules now allow him to start with some amount of money, he will be able to make it to open the nth box. Implement the function `int start_amt(int n)` that takes the number of boxes to be opened as the input, and returns the minimum starting amount of money that is required to open at least n boxes.

In [ ]:
```
double cost(double surcharge, int weight)
{
    int total = 0, min = 0;
    for(int i = 0; i < n; i++)
    {
        total += open_box();
        if(total < min) min = total;
    }
    return -min;
}
```

Question 3: Treasure Hunt

Captain Spack Jarrow is looking for treasure buried along a beach. Digging at any spot on the beach will reveal a clue which indicates how many steps forwards or backwards to dig next, which may reveal more of such clues or hopefully, the treasure! We can model the beach as a vector of integers, which each element being the clue, i.e., the number of steps to move forwards or backwards. Positive integers indicate steps forward while negative integers indicate steps backwards. Zero (0) will indicate the treasure! Now the beach is on an island and loops around on itself endlessly. Thus, the vector representation of the beach wraps around, i.e., taking a step off either ends will bring you to the opposing end. For example, consider a model of the beach shown below:

In [13]:
```
from IPython.display import Image
```



If Spack starts digging at index 1, he will get a clue to move 7 steps forward (to the right). Digging there will result in going 6 spaces forward, wrapping around to index 3, followed by 3 steps forward, and finally 2 steps backward to find the treasure, after a total of 5 digs.

A.

Implement the function `int num_digs(vector beach, int start)` , that takes a vector of int that models the beach, and the starting index as inputs. It returns the number of times that Spack must dig before uncovering the treasure. You may assume that the starting index will always eventually lead to the treasure.

In [ ]:
```
int num_digs(vector<int> beach, int start)
{
    int num = 1;
    while(beach[i] != 0)
    {
        num ++;
        start = (start + beach[start] + beach.size()) % beach.size();
    }
    return digs;
}
```

B.

Now it might be possible that the treasure will never be found. For example, if Spack started at index 0 in the example above, he will never find the treasure. Without creating any new strings, arrays or vectors, how would you modify your implementation of `num_digs` such that it returns -1 if the treasure will never be found? You may either describe in words or rewrite the function.

In [ ]:
```
int num_digs(vector<int> beach, int start)
{
    int num = 1;
    while(beach[i] != 0)
    {
        if(num > beach.size()) return -1;
        num ++;
        start = (start + beach[start] + beach.size()) % beach.size();
    }
    return digs;
}
```

C.

Implement the function `double avg_digs(vector beach)` that takes a vector of int that models the beach as input, and returns the expected (or average) number of digs Spack will take if he started at any random index. Assume that every starting index will lead to the treasure.

In [ ]:
```
double avg_digs(vector<int> beach)
{
    int total = 0;
    for (int i = 0; i < beach.size(); i++)
    {
        total += num_digs(beach, i);
    }
    return (double)total / beach.size();
}
```

Question 5: Computer Organization & Data Representation

Answer all parts in this question with 1–2 words or one number. Please write your answer clearly.

A. If a 24-bit (i.e. 3-byte) RGB value is given as $1A2B3C_{16}$, what is the value for Green color in base 10?

Each colour can be splitted into:

R: $1A_{16}$
G: $2B_{16}$
B: $3C_{16}$

Hence, Green = $2B_{16}$ = $2 * 16^1 + 11 * 16^0$ = 43

B.The GCC/G++ compiler suite actually contain multiple sub-components. Typically a user program pass through the sub-components in the following order:

Compiler -> Assembler -> Linker

C. A high level language instruction like i = i+1; may be separated in the following processor instructions:

1. Load i from memory into Register R
2. add R by 1
3. store R back into memory

Question 6: Cache & Operating System

Answer all parts in this question with one word or one number. Please write your answer clearly.

A. Evaluate whether the following statements are true or false.

i) "With multi-tasking, there are more than one process executing on the processor (single core) at any point in time."

False

ii) "With multi-tasking, there are more than one process residing in the memory (RAM) at any point in time."

True

B. In modern computer systems, information (e.g. data, instruction) may pass through the following storage during execution:

Hard Disk -> RAM -> Cache -> Register

(From large to small)

C. Given the following memory block accesses:

7, 2, 2, 23, 7, 16, 23, 16

Give the number of cache misses with the following cache configuration.

i) Fully Associative Cache with two cache blocks. The oldest block is replaced when necessary

Block 0 is less recently used, Block 1 is more recently used

0	1	Misses	Notes
7		1	Allocate block
7	**2**	1	Allocate block
7	2		Matched 2, 2 is most recently used
23	2	1	Allocate 23
23	**7**	1	Allocate 7
16	7	1	Allocate 16
16	**23**	1	Allocate 23
23	16		Matched 16, 16 is most recently used

Ans: 6 misses

ii) Direct Map Cache with two cache blocks.

Block 0 is less recently used, Block 1 is more recently used

0	1	Misses	Notes
	7	1	7 % 2 = 1
2	7	1	2 % 2 = 0
2	7		Matched 2
2	**23**	1	23 % 2 = 1
2	**7**	1	7 % 2 = 1
16	7	1	16 % 2 = 0
16	**23**	1	23 % 2 = 1
16	23		Matched 16

Ans: 6 misses

Question 7: Database

A .Evaluate whether the following statements are true or false.

i. "Atomicity means multiple transactions can be handled properly."

False

ii. "If the end result of executing multiple database transactions concurrently is not the same as a particular sequential execution of those same transactions, then that database violates the isolation property."

False

C. Suppose we have a database table created for car in the Singapore market using the following SQL statement:

In [ ]:
```
CREATE TABLE Car (Model varchar(40),
                  Brand varchar(20),
                        EngineCC int, HorsePower int, Price float);
```

Complete the following SQL queries according to the given description:

i) List all Category A car for the “Hyundai” brand. According to LTA’s definition, Category A cars’ “engine capacity should not exceed 1,600 cc” and “engine power should not exceed 130 horse-power”.

In [ ]:
```
SELECT *
FROM Car
WHERE Brand="Hyundai"
  AND EngineCC <= 1600
  AND HorsePower <= 130;
```

ii) List Category B cars in ascending order according to price. Cars that failed Category A criteria are classified as category B cars.

In [ ]:
```
SELECT *
FROM Car
WHERE EngineCC > 1600
  AND HorsePower > 130
ORDER BY Price ASC;
```

— END OF PAPER —