TIC1001—Introduction to Computing and Programming
National University of Singapore

Final Exam - AY19/20

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 = "pineapple pen";

int i = 0, j = s.size() - 1;

while (i < j)
{
    s[i] = s[j];
    s[j] = s[i];
    i += 1;
    j -= 1;
}
cout << s << endl;
```

Initialize string s = "pineapple pen"

i = 0, j = 12

0.
while(i < j)
s[0] = s[12] -> nineapple pen
s[12] = s[0] -> nineapple pen
i = 1, j = 11

1.
while(i < j)
s[1] = s[11] -> neneapple pen
s[11] = s[1] -> neneapple pen
i = 2, j = 10

2.
while(i < j)
s[2] = s[10] -> nepeapple pen
s[10] = s[2] -> nepeapple pen
i = 3, j = 9

3.
while(i < j)
s[3] = s[9] -> nep apple pen
s[9] = s[3] -> nep apple pen
i = 4, j = 8

4.
while(i < j)
s[4] = s[8] -> nep epple pen
s[8] = s[4] -> nep epple pen
i = 5, j = 7

5.
while(i < j)
s[5] = s[7] -> nep elple pen
s[7] = s[5] -> nep elple pen
i = 6, j = 6

6.
i != j, exit out from while loop.

Ans: nep elple pen

B.

In [ ]:

```
for (int i = 0; true; i++)
{
    cout << i << " ";
    if (i < 7)
    {
        i += 2;
    }
    else if (i % 3 == 0)
    {
        i /= 2;
        continue;
    }
    else
    break;
}
```

i = 0, **cout: 0**
i < 7, i += 2 = 2

i = 3, **cout: 3**
i < 7, i += 2 = 5

i = 6, **cout: 6**
i < 7, i +=2 = 8

i = 9, **cout: 9**
i % 3 == 0, i /= 2 = 4
i = 5, **cout: 5**
i < 7, i +=2 = 7

i = 8, **cout: 8**
Break, exit from loop.

Ans: 0 3 6 9 5 8

C.

In [ ]:

```
double f(int &x, int y)
{
    x /= y;
    cout << x << " ";
    return x;
}

int g(double x, double &y)
{
    y /= x;
    cout << y << " ";
    return y;
}

int main()
{
    int x = 22;
    double y = 5;
    cout << g(f(x, y), y);
}
```

Initialize
x = 22, y = 5
g(f(22,5),5)

f(&x,y) -> f(&22,5)
x = 22/5 = 4
return 4
cout: 4
x = 4, y = 5

g(4,&5)
y = 5/4 = 1.25 *(5 is double)*
return 1 *(return type is int)*
cout: 1.25
x = 4, y = 1.25

cout: 1

Ans: 4 1.25 1

D.

In [ ]:

```
int a[] = {5, 6, 10, 4, 1, 8, -1, 2, 3, 7};

int x = 0;

while (a[x] > 0)
{
    x = (x + a[x]) % 10;
    cout << x;
}
```

x = 0

a[0] = 5 > 0,
x = (0 + 5) % 10 = 5
cout: 5

a[5] = 8 > 0,
x = (5 + 8) % 10 = 3
cout: 3

a[3] = 4 > 0,
x = (3 + 4) % 10 = 7
cout: 7

a[7] = 2 > 0,
x = (7 + 2) % 10 = 9
cout: 9

a[9] = 7 > 0
x = (9 + 7) % 10 = 6
cout: 6

a[6] = -1 < 0
Exit loop.

cout: 5 3 7 9 6

Question 2: Airline Baggage

A.

A particular budget airline has the following pricing scheme for checked baggage:

Weight	Cost
15kg	20
25kg	40
40kg	90
>40kg	+5 per kg

The cost is computed part-thereof, so 17 kg baggage will cost 40. Baggage over 40 kg will cost 90 plus 5 for every excess kg above 40.

[Warm up]
Implement the function `int cost(int weight)` that takes as input the weight of the baggage, and returns the cost of the baggage following the pricing scheme above.

In [ ]:

```
int cost(int weight)
{
    int cost;
    if(weight <= 15) cost = 20;
    else if (weight > 15 && weight <= 25) cost = 40;
    else if (weight > 25 && weight <= 40) cost = 90;
    else cost = ((weight - 40) * 5) + 90;
}
```

B.

Another airline uses a simpler pricing scheme. It simply charges a fix price of c dollars for every n kg or part thereof. For example, if c = 10 and n = 5kg, then a 17 kg baggage will cost 40.

Implement the function `int cost(int c, int n, int weight)` that takes as input the price, the kg part thereof, and the weight of the baggage, and returns the cost of the baggage.

the cost of the baggage.

In [ ]:

```
int cost(int c, int n, int weight)
{
    int cost = 0;
    while (weight > 0)
    {
        cost += c;
        weight -= 5;
    }
}
```

C.

Yet another airline has another pricing scheme. It charges 20 for the first 20 kg, and an extra 5 for the next 5 kg or part thereof. Subsequently, for every 5 kg, it increases the extra cost by a percentage surcharge.

For example, if the surcharge is 50%, (i.e., 0:5), then after charging 25 for the first 25 kg, the next 5 kg will cost 5 x 1:5 = 7:50. Another 5 kg will cost 7:5 x 1:5 = 11:25, ad infinitum.

Implement the function `double cost(double surcharge, int weight)` which takes in the surcharge (as a decimal ratio, e.g. 50% is 0.5) and the weight of the baggage, and returns the cost of the baggage. You should use implement using computation and not use a formula.

In [ ]:

```
double cost(double surcharge, int weight)
{
    double cost;
    if (weight <= 20) cost = 20;
    else if (weight > 20 && weight <= 25) cost = 25;
    else
    {
        cost = 25;
        weight -= 25;
        double base = 5;
        while( weight > 0)
        {
            base = base * (1 + surcharge);
            cost += base;
            weight -= 5;
        }
        return cost;
    }
}
```

Question 3: Scrabble

[INSTRUCTIONS] Use of any C++ STL functions is not allowed unless otherwise stated.Functions defined in other parts should be used whenever appropriate.

The game of Scrabble is played using tiles with letters marked on them. Players then attempt to form words on the board using the tiles which they have on hand. For this question, a set of tiles will be modelled as a C++ vector of char, where each char represents the letter of the tile. You may assume all tiles and letters given are all upper case.

A.

The function `int find(vector tiles, char letter)` takes in a set of tiles,and a letter. If the set of tiles contains the letter, `find` returns an index of the vector that matches it. Otherwise, `find` returns -1.

Provide an implementation of the function `find` .

In [ ]:

```
int find(vector<char> tiles, char letter)
{
    for(int i = 0; i < tiles.size(); i++)
    {
        if(tiles[i] == letter)
            return i;
    }
    return -1;
}
```

B.

When a tile is placed on the board, it will be removed from the player's hand. The function remove takes as input a set of tiles and a letter. It modifies the input tiles by removing a single tile matching the given letter.

Provide an implementation for the function `remove` . You are to decide on the appropriate return type and parameters for the function.

In [ ]:

```
void remove (vector<char> tiles, char letter)
{
    tiles.erase(tile.begin() + find(tiles,letter));
}
```

C.

Players needs to know if a particular word can be formed from their set of tiles. The function `bool can_form(vector tiles, string word)` takes a set of tiles and a string as input, and returns true if all characters of the word can be uniquely matched to the letters in the set of tiles. I.e., duplicate characters in word requires just as many duplicate tiles in the players hand.

For example, if a player has the tiles with letters: **A, H, O, X, P, Y, O** on hand, the word **"HAPPY"** cannot be formed because there is only one **P** in the set of tiles. But the word **"HOOP"** can be formed because there are two O's in the set of tiles.

Provide an implementation for the function `can_form`.

In [ ]:

```
bool can_form(vector<char> tiles, string word)
{
    for(int i = 0; i < word.size(); i++)
    {
        int pos = find(tiles, word[i]);
        if(pos == -1) return false;
        remove (tiles, word[i]);
    }
    return true;
}
```

Question 4: Caching, OS & Database

A. Determine which of the following statements are true or false with regards

to running several processes on a multi-tasking operating system.

i) Only one process can be executing on a multi-core CPU at any point in time.

False.

More than one process can be executed.

ii. When execution switches to a new process, all data from the old process is cleared from the RAM.

False.

Not necessarily.

iii. Different processes can share data and communicate with each other by writing to and reading from the same memory address.

False.

iv. When a process incurs a page fault, another process might take over to execute on the CPU while the operating system services the page fault.

True.

B. Given the following memory block accesses:

7, 2, 2, 23, 7, 16, 23, 21, 7, 2

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

i. Fully Associative Cache with four cache blocks. The least recently used block is replaced when necessary.

Block 0 is less recently used, Block 3 is more recently used

0	1	2	3	Misses	Notes
7				1	Allocate block
7	**2**			1	Allocate block
7	2				Matched, 2 is most recently used
7	2	**23**		1	Allocate block
2	23	7			Matched, 7 is most recently used
2	23	7	**16**	1	Allocate block
2	7	16	23		Matched, 23 is most recently used
21	7	16	23	1	Allocate block, replace least recently used
21	16	23	7		Matched, 7 is most recently used
2	16	23	7	1	Allocate block, replace least recently used

Ans: 6 misses

ii. Direct Map Cache with four cache blocks. Memory blocks in the form 4n+k are mapped to cache block k.

0	1	2	3	Misses	Notes
			7	1	7 % 4 = 3
		2		1	2 % 4 = 2
		2			2 % 4 = 2, matched
		2	**23**	1	23 % 4 = 3
		2	**7**	1	7 % 4 = 3
16		2	7	1	16 % 4 = 0
16		2	**23**	1	23 % 4 = 3
16	**21**	2	23	1	21 % 4 = 1
16	21	2	**7**	1	7 % 4 = 3
16	21	2	7		2 % 4 = 2, matched

Ans: 8 misses

C. Consider the contents of the FLIGHT table in an SQL database:

Airline	Source	Destination	Plane
BA	BKK	LHR	777
BA	JFK	LHR	777
BA	SIN	LHR	744
KE	ICN	JFK	388
KE	ICN	LAX	77W
KE	LHR	ICN	77W
KE	SIN	ICN	773
SQ	BKK	SIN	773
SQ	FRA	SIN	77W
SQ	JFK	FRA	338
SQ	SFO	ICN	77W
UA	FRA	JFK	744
UA	JFK	LAX	757
UA	LHR	SFO	777

Give the simplest SQL query for each of the corresponding result:

Result:

BA	777
SQ	388
UA	757

In [ ]:

```
SELECT
        Airline,
        Plane
    FROM
        FLIGHT
    WHERE
        Source = "JFK"
```

Result:

BA	SIN	LHR	744
KE	SIN	ICN	773
SQ	BKK	SIN	773
SQ	FRA	SIN	77W

In [ ]:

```
SELECT
        *
    FROM
        FLIGHT
    WHERE
        Source = "SIN"
        OR Destination = "SIN"
```

— END OF PAPER —