TIC1001—Introduction to Computing and Programming
National University of Singapore

Midterm Test - AY17/18

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 [1]:
```
#include <iostream>
using namespace std;
```

A.

In [ ]:
```
int square(double x)
{
    return x * x;
}
printf("%d", square(1.5));
```

0. Call `square(x) -> square(1.5)`

return 1.5 * 1.5 = 2.25

However return data type is int, return 2.

Print"2"

Ans: 2

B.

In [ ]:
```
int x = 2;
int square(int x)
{
    return x * x;
}
printf("%d", square(square(3)));
```

Initialize x = 2

0.Call `square(x) -> square(3)`

return 3 * 3 = 9, return 9

1.Call `square(x) -> square(9)`

return 9 * 9 = 81, return 81

Ans: 81

C.

In [ ]:
```
int x = 5;
int y = 3;
double foo(int x, int y)
{
    return x/y;
}
printf("%f", foo(y, x));
```

Initialize x = 5, y = 3

Call `foo(y,x) -> foo(3,5)` return 3 / 5 = 0

Print"0.000000"

D.

In [ ]:
```
int foo(int x, double y)
{
    printf("%.2f ", y/x);
    return y/x;
}
foo(foo(4, 10), 15);
```

0.Call `foo(4,10)`

10 / 4 = 2.50

Print"2.50", return 2

1.Call `foo(2,15)`

15 / 2 = 7.50

Print"7.50", return 7

Ans: 2.50 7.50

E.

In [ ]:

```c
int i;
for (i = 0; i < 10; ++i)
{
    if (i % 2 == 0)
    {
        ++i;
    }
    if (i % 3 == 0)
    {
        --i;
    }
    if (i % 5 == 0)
    {
        break;
    }
}
printf("%d", i);
```

When i = 0,
0 % 2 = 0, will pass `if (i % 2 == 0)`
Increment i + 1 = 1
Continue to next If Statement
1 % 3 = 1, will not pass `if (i % 3 == 0)`
Continue to next If Statement
1 % 5 = 1, will not pass `if (i % 5 == 0)`
Continue to loop, increment i + 1, i = 2

When i = 2,
2 % 2 = 0, will pass `if (i % 2 == 0)`
Increment i + 1 = 3
Continue to next If Statement
3 % 3 = 0, will pass `if (i % 3 == 0)`
Decrement i - 1 = 2
Continue to next If Statement
2 % 5 = 2, will not pass `if (i % 5 == 0)`
Continue to loop, increment i + 1, i = 3

When i = 3,
3 % 2 = 1, will not pass `if (i % 2 == 0)`
Continue to next If Statement
3 % 3 = 0, will pass `if (i % 3 == 0)`
Decrement i - 1 = 2
Continue to next If Statement
2 % 5 = 2, will not pass `if (i % 5 == 0)`
Continue to loop, increment i + 1, i = 3

When i = 3,
3 % 2 = 1, will not pass `if (i % 2 == 0)`
Continue to next If Statement
3 % 3 = 0, will pass if (i % 3 == 0)
Decrement i - 1 = 2
Continue to next If Statement
2 % 5 = 2, will not pass `if (i % 5 == 0)`
Continue to loop, increment i + 1, i = 3

Ans: i will stuck at 3, infinite loop

F.

In [ ]:

```c
int a = 3;
int f(int a)
{
    a = 7;
    return a;
}
int g(int a)
{
    return f(a) + a;
}
printf("%d", g(a+2) + f(a-1));
```

Initialize a = 3

Call `g(3+2) -> g(5)`
Call `f(a)` -> 7
Return 7 + 5 = 12

Call `f(a-1) -> f(2)`
Return 7

12 + 7 = 19,print"19"

G.

In [ ]:
```c
inti=20,j=0;
while (i >= j)
{
    --i;
    if (i < 3*j)
    {
        continue;
    }
    j += 2;
    printf("%d ", j);
}
printf("%d", i);
```

Initialize i = 20, j = 0

20 >= 0, enter while loop.
Decrement i - 1, i = 19
19 > 3 *0, will not pass* `if (i < 3    j)</code>`

`Increment j + 2, j = 2`

`Print"2 "`

`19 >= 2, enter while loop.`

`Decrement i - 1, i = 18`

`18 > 3` *2, will not pass* `if (i < 3    j)</code>`

`Increment j + 2, j = 4`

`Print"4 "`

`18 >= 4, enter while loop.`

`Decrement i - 1, i = 17`

`17 > 3` *4, will not pass* `if (i < 3    j)</code>`

`Increment j + 2, j = 6`

`Print"6"`

`17 >= 6, enter while loop.`

`Decrement i - 1, i = 16`

`16 < 3` *6, will pass* `if (i < 3    j)</code>`

`Continue to loop`

`16 >= 6, enter while loop.`

`Decrement i - 1, i = 15`

`15 < 3` *6, will pass* `if (i < 3    j)</code>`

`Continue to loop`

`15 >= 6, enter while loop.`

`Decrement i - 1, i = 14`

`14 < 3` *6, will pass* `if (i < 3    j)</code>`

`Continue to loop`

.....i will continue to decrement by 1 in every loop.

5 >= 6, will not enter while loop.

Print "5"

Ans: 2 4 6 5

In [ ]:

```
int x = 0;
int f(int *x)
{
    if (*x > 0)
    {
        *x = -*x;
    }
    else
    {
        *x += 2;
    }
    return *x;
}

x = f(&x) + f(&x);
printf("%d", x);
```

Initialize x = 0

0.Call f(&x)

* x = 0, will not pass <code>If (* x > 0)</code>. Enter Else statement<br>
* x + 2 = 0 + 2 = 2, return 2

1.Call f(&x)

* x = 2, will pass <code>If (* x > 0)</code>.<br>
* x = - 2, return -2

x = 2 + (-2) = 0

Print"0"

Ans: 0

— END OF PAPER —