Solutions - Homework 2

(Due date: Oct. 5th)

PROBLEM 1 (20 PTS)

- Refer to Activity 2 in the *High-Performance Embedded Programming with the Intel® Atom™ platform → Tutorial 2*
  - ✓ Activity 2 – Image Convolution in C: Execute the application so that it generates the `iss.bof` file. Provide a screenshot of the execution in the Terminal. (10 pts)

- Based, on the completion of the Activity 2, answer the following questions:
  - ✓ Input image pixels ∈ [0,255] (integer range). For a pixel, what data type should be used (mark the correct answer) ?

    `char` `double` `int` ~~`unsigned char`~~

  - ✓ Why is it that the output image values might fall outside the [0,255] integer range? (3 pts)
    When multiplying numbers between [0 255] with the kernel that has numbers -1 and 4, it is possible that some numbers end up outside the range [0, 255]. Example: 15*-1 + 10*-1 + 5*5 + 10*-1 + 15*-1 = -45.

  - ✓ In the code of Activity 2, what data type are the output image values (the ones written on the `.bof` file) assigned?

    `char` `double` ~~`int`~~ `unsigned char`

  - ✓ Why is the size of the output `.bof` file 4 times as much as the size of the input `.bif` file? (2 pts)
    `.bif` file: Each element is 8-bits wide (`unsigned char`)
    `.bof` file: Each element is 32-bits wide (`int`)

  - ✓ For proper displaying, it is customary for grayscale image pixels to be 8-bit unsigned integers. If the output matrix values fall outside the [0,255] range, we can perform saturation in order to convert the element of the output matrix to 8-bit unsigned integers. This way the output matrix can be properly displayed on a screen. Complete the following table:

Output Matrix values (generated by the .c code):	Output matrix values converted to 8-bit unsigned integers:
256	255
37	37
-255	0
-128	0

- For a real-valued kernel, we would need to re-write the code to generate a real-valued output matrix. However, for proper displaying, the output matrix values would need to be converted to 8-bit unsigned integers via rounding and saturation. Complete the following table. (2 pts)

Output Matrix values (generated by a .c code):	Output matrix values converted to 8-bit unsigned integers:
278.35	255
-256.78	0
-128.59	0
78.25	78

PROBLEM 2 (20 PTS)

- In the following code snippet, a class `Circle` is defined. Then in `main()`, we use the class to compute the perimeter of a circle given the radius. Two options are provided.

```
using namespace std;
class Circle {
private:
  float radius;
public:
  Circle () {}
  Circle (float ra): radius(ra) {}
  void compute_perimeter () {
     float perimeter = 3.14*2*radius;
     cout << "Perimeter is :" << perimeter << endl;
  }
};
```

Option 1	Option 2

`int main() {`
`  Circle C(3);`
`  C.compute_perimeter();`
`  return 0;`
`}`
	`int main() {`
	`  Circle C;`
	`  C.radius = 3;`
	`  C.compute_perimeter();`
	`  return 0;`
	`}`

- ✓ Option 1: Syntax-wise, is it correct or incorrect? Why?
  This is the correct, as `Circle C(3)` properly initializes `C.radius`.

- ✓ Option 2: Syntax-wise, is it correct of incorrect? Why?
  This is incorrect, as we are incorrectly trying to assign `C.radius` in `main()`. `radius` is private and cannot be accessed outside the class.

PROBLEM 3 (20 PTS)

- In the following code snippet, a class `Sample` is defined. Then in `main()`, we create two objects and perform associated operations.

```
using namespace std;
class Sample {
public:
   int x, y, z, s;
   Sample (): x(2), y(3), z(4) {}
   Sample (int xa, int ya, int za): x(xa), y(ya), z(za) {}

   int operation() {
     s = x*y*z;
     return s; }

   int operation(int offset) {
     s = x*y*z + offset;
     return s; }
};

int main() {
  int result_1, result_2;

  Sample S1;
  Sample S2(3,4,5);
  result_1 = S1.operation();
  result_2 = S2.operation(8);

  cout << "Result (S1): " << result_1 << endl;
  cout << "Result (S2): " << result_2 << endl;

  return 0;
}
```

- ✓ Provide the result of the Program Output: (10 pts)
  `Result (S1): 24`
  `Result (S2): 68`

- ✓ If `S1` had been declared as `Sample S1(4,5,6)` (instead of `Sample S1`), what would be the program output on the first printed line:
  - `Result (S1): 120`

- ✓ If `S2` had been declared as `Sample S2` (instead of `Sample S2(3,4,5)`), what would be the program output on the second printed line:
  - `Result (S2): 32`

PROBLEM 4 (25 PTS)

- In the following code snippet, a class `TestFunctor` is defined. Then in `main()`, we use the class to compute the perimeter of a circle given the radius. Three options are provided.

```
class Test {
public:
   Test (): x(1), y(1) {}
   Test (int xt, int yt): x(xt), y(yt) {}

   int operator() (int zt) {
      int b;

      z = zt;
      b = x*x + y*y + z*z;
      return b;
   }

private:
   int x;
   int y;
   int z;
};
```

Option 1	Option 2	Option 3

`int main() {`
`  int r;`
`  Test myfunctor;`
`  myfunctor.x =4; myfunctor.y =6;`
`  r = myfunctor(7);`
`  cout << "r: " << r << "\n";`
`  return 0;`
`}`
	`int main() {`
	`  int r;`
	`  Test myfunctor (4,6);`
	`  r = myfunctor(7);`
	`  cout << "r: " << r << "\n";`
	`  return 0;`
	`}`
		`int main() {`
		`  int r;`
		`  Test myfunctor;`
		`  r = myfunctor(7);`
		`  cout << "r: " << r << "\n";`
		`  return 0;`
		`}`

- ✓ Which option(s) is (are) syntactically incorrect? ~~Option 1~~ Option 2 Option 3
  - For the incorrect option(s), why are they incorrect? (7 pts)
    Option 1 is incorrect because it is trying to access `myfunctor.x` and `myfunctor.y` when `x` and `y` are private.
  - For the correct option(s), what is the value of `r`? (10 pts)
    Option 2: `r = 101`
    Option 3: `r = 51`

PROBLEM 5 (15 PTS)

- Refer to the Activity 4 in the *High-Performance Embedded Programming with the Intel® Atom™ platform → Tutorial 2*
  - ✓ Activity 4 – Neuron: Execute the application. Provide a screenshot of the execution in the Terminal. (10 pts)

- Based, on the completion of the Activity 4, answer the following questions:
  - ✓ Is it correct to include the following line in `main()`? Why or why not?
    - `cout << "AN.a: " << AN.a << endl;`

    No, it is not correct. Because `a` is private and cannot be accessed outside the class
  - ✓ To allocate memory in `main()` for `AN.a`, could have done the following? Why or why not? (3 pts)
    - `AN.a (double *) calloc (NI, sizeof(double));`

    No. Because `a` is private and cannot be accessed (or allocated) outside the class