Week 4 Quiz Answer

Date	03/03/2023	Time	15 mins

1. *(15 pts each, 60 pts total)* Answer the following questions.

   (a) What is a double pointer?
   → *Double pointer is a pointer that points to another pointer.*

   (b) Write code that dynamically allocates memory for a 2-dimensional array (`arr_2d`) with `m` rows and `n` columns. *Note: All the memories of this array must be dynamically allocated.*
   →
   ```
   int** arr_2d = new int*[m];
   for (int i = 0; i < m; i++)
       arr_2d[i] = new int[n];
   ```

   (c) Write code that deallocates all the memories of the above `arr_2d`.
   →
   ```
   for (int i = 0; i < m; i++)
       delete[] arr_2d[i];
   delete[] arr_2d;
   arr_2d = NULL;
   ```

   (d) Why do we need pointers to functions in C++?
   → *We can use pointers to functions for passing functions as arguments to other functions.*

2. *(10 pts each, 20 pts total)* Given the following C++ code. Fulfill the following requirements.

```
char* strs[] = {"Hello", "22CLC7", "!"};
char** ptr = &strs[1];
```

   (a) Write a statement that uses `cout` to display the first string in `strs` (`"Hello"`) via `ptr` (do not use `strs`).
   → `cout << *(ptr - 1);`

   Explanation: `ptr` *is a double pointer to the address of the second element of* `strs`.
   → `(ptr - 1)` *will point to the address of the first element of* `strs` *(*`strs[0]`*, a pointer to the address of string* `"Hello"`*). So* `*(ptr - 1)` *will give us the address of string* `"Hello"`*, and we only need to use* `cout << *(ptr - 1)` *to display the string.*

   (b) Write a statement that uses `cout` to display **the address** of the above string (`"Hello"`) via `ptr` (do not use `strs`).
   → `cout << (int*)(*(ptr - 1));`

   Note: If we apply address-of operator to `*(ptr - 1)` i.e. `&(*(ptr - 1))`*, the result will be equal to* `(ptr - 1)`*.* → *It will* ***return the address of the first element*** *of the* `strs` ***NOT the address of the string*** `"Hello"`.

3. *(20 pts)* In any case, what is the output from the program below? Explain your answer.

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

bool f1(int a, int b) {
    return a < b;
}

bool f2(int a, int b) {
    return a > b;
}

int do_sth(bool (*fp)(int, int), int a, int b, int c) {
    int temp = a;
    if (fp(temp, b))
        temp = b;
    if (fp(temp, c))
        temp = c;
    return temp;
}

int main() {
    int a = 5, b = 10, c = 15, d = 20;

    cout << do_sth(f1, a, b, d) << " " << do_sth(f2, c, b, d);

    return 0;
}
```

→ *The output is* `20 10`.

Explanation:

- `f1` *checks whether a is less than b.*
- `f2` *checks whether a is greater than b.*
- `do_sth` *will return the minimum element of the three elements (a, b, c) if we pass the* `f2` *function to it (*`if (temp > b) temp = b;`*). Likewise to finding the maximum element.*

END