Q1: Object Oriented Programming Basics - Attributes, Methods, Inheritance, Access Specifiers

1. Create a class “Employee”. Each employee has a (public) unique integer attribute “id” and a void member function “work” which should print “I am Employee <id here> and I am working”. (The constructor of Employee should initialise the ID)

2. Add a *secret* int attribute “password” to the Employee class and also create a function login which takes an integer argument, compares it with the secret password and prints “Login Successful” or “Unsuccessful” to the user.

3. Create two classes “Developer” and “Manager” which *inherit* from the class “Employee”. The behaviour of the Developer class should be the same as “Employee”, but the Manager has an additional int “manager id” attribute and the “work()” function of Manager should print “I am Manager <manager-id> and I am working”. The login function should work the same.

4. The Manager has another power that they are able to change their own password. Create a function change_password which takes the old_password and the new_password and sets the password attribute to the new password. *(To illustrate private vs protected attributes)*

Q2: Pointers/Pass by Reference Intro

1. Create a function which takes two integer pointers and swaps their values.
2. Do the same Q without using pointers.

Q3: Stack vs Heap Allocation Example

Make the students write this code and ask them to explain the output (maybe ask them to read the relevant sections of the C/C++ module)

```
#include<iostream>
using namespace std;
class A {
public:
    int id;
    A(int i) {
        id = i;
        cout << "Created object with ID: " << id << endl;
    }
    ~A() {
        cout << "Destroyed object with ID: " << id << endl;
    }
};
int main() {
    A a1(3);
    A* a2 = new A(5);
    return 0;
}
```

What will be the output of the above code you expect on the terminal? Briefly explain.