LAB MANUAL 10-B

DOUBLE POINTERS

WHAT ARE DOUBLE POINTERS?

A double pointer is a variable that points to another pointer which in turn, points to an object in memory.

Examples:

```
int value = 100;
int *value_ptr = &value;
int **value_double_ptr = &value_ptr;

printf("Value: %d\n", value);
printf("Pointer to value: %d\n", *value_ptr);
printf("Double pointer to value: %d\n", **value_double_ptr);
```

Output:

```
Value: 100
Pointer to value: 100
Doublue pointer to value: 100
```

When dereferencing a double pointer, we do not get the final object, but what is expected: a pointer that must be dereferenced one more time to retrieve the final value. It would be similar to the code below.

```
int *ptr = *value_double_ptr;
int final_value = *ptr;
```

Tasks

1. Correct errors if any and Print Output

a)

```
#include<iostream>
using namespace std;
main()
{
    float num = 10;
    float *pt1 = &num;
    float **pt2 = &pt1;
    cout<<" address of num =  "<<pt1<<endl;
    cout<<" address stored by pt1 = "<<&num<<endl;
    cout<<" value pointed by pt1 = "<<*pt1<<endl;
    cout<<" address of pt1 = "<<pt2<<endl;
    cout<<" address stored by pt2 = "<<&pt1<<endl;
    cout<<" value of num  = "<<**pt2;
}
```

b)

```
#include<iostream>
using namespace std;
main()
{
    char *p1;
    int a = 8;
    p1 = &a;
    cout<<p1;
}
```

c)

```
#include<iostream>
using namespace std;
main()
{
    int a =5;
    int *p = &a;
    cout<<++*p<<endl;
    cout<<*p++;
}
```

d)

```
#include<iostream>
using namespace std;
main()
{
    int a =5;
    int *p = &a;
    cout<<p<<endl;
    cout<<++*p<<endl;
    cout<<*p++;
    cout<<p;
}
```