Name: Muhammad Zohaib khan

Reg# BSE203003

Subject: OOP LAB

Instructor: Sir Qaisar Manzoor

Date:20-3-2021

Practice Task 1

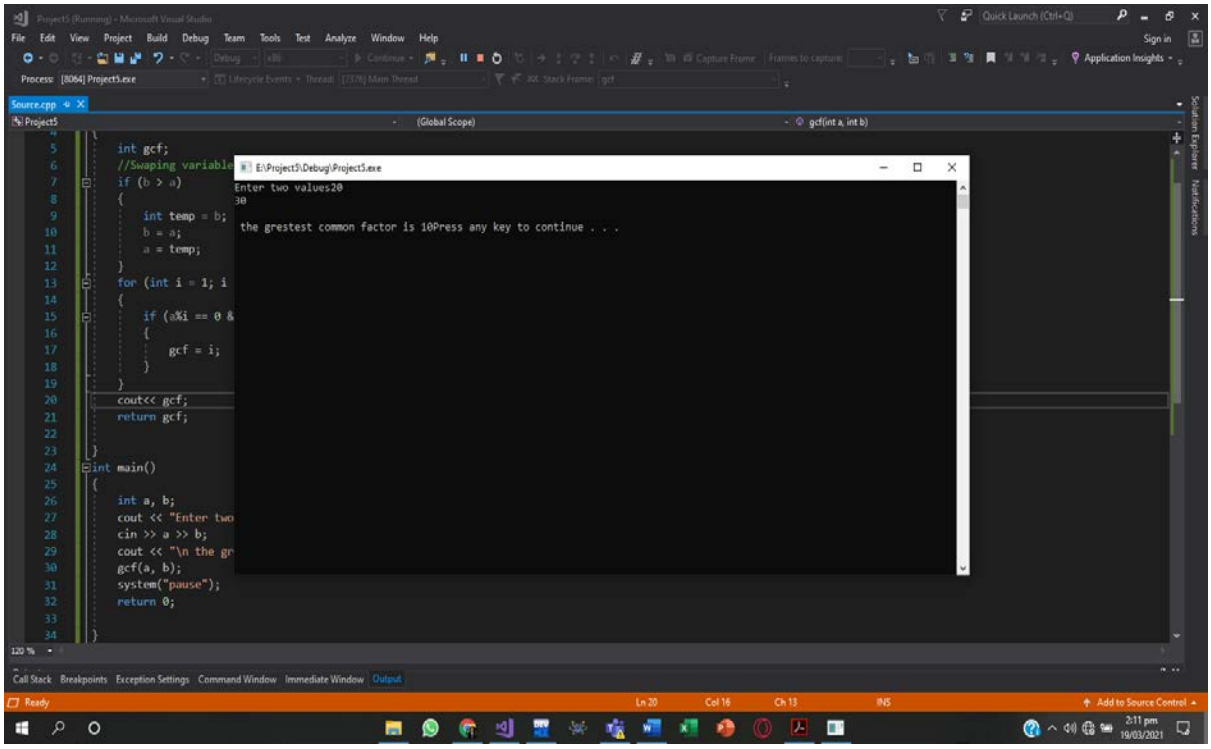
Write a program to find the greatest common factor if the two numbers are provided in the main function. Use pass by value mechanism to compute the greatest common factor and return the result back to the main function then display the result in the main function.

Code:

```
#include<iostream>
using namespace std;
int gcf(int a, int b)
{
	int gcf;
	//Swaping variables
	if (b > a)
	{
		int temp = b;
		b = a;
		a = temp;
	}
	for (int i = 1; i <= b; ++i)
	{
		if (a%i == 0 && b%i == 0)
		{
			gcf = i;
		}
	}
	cout<< gcf;
	return gcf;

}
int main()
{
	int a, b;
	cout << "Enter two values";
	cin >> a >> b;
	cout << "\n the grestest common factor is ";
	gcf(a, b);
	system("pause");
	return 0;

}
```



Practice Task 2

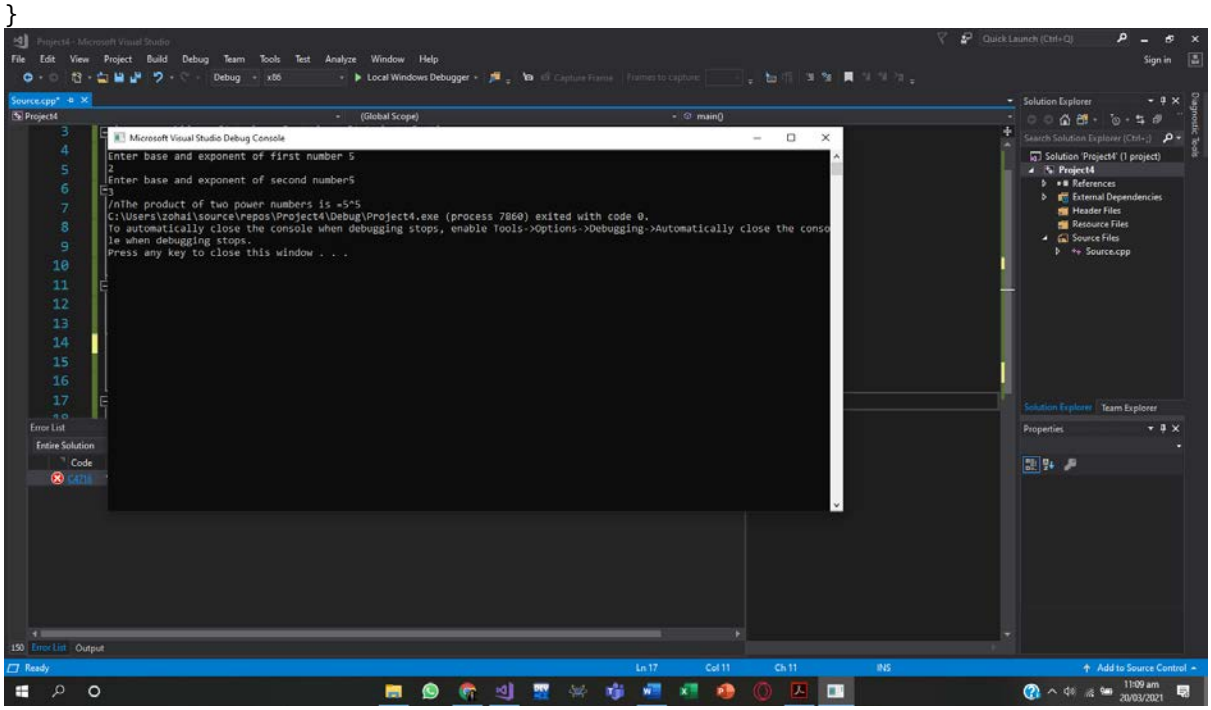
Write a program to find the product of 2 power numbers if the base and the exponent of these numbers are provided in the main function, also check that the base of both the power number are the same, as the product of 2 power numbers can only be calculated if their base is the same. Use pass by reference mechanism to compute the product of these power numbers then display the result in main function.

Code:

```
#include<iostream>
using namespace std;
int prod(int &b1,int &e1,int &b2,int &e2)
{
	int e3;
	if (b1 == b2)
	{
		e3 = e1 + e2;
		cout << "/nThe product of two power numbers is =" <<b1<<"^"<< e3;
}
	else {
		cout << "/nThe base of two numbers are not same ";
	}
	return e3;

}
int main()
{
	int b1, e1;
	int b2, e2;
	cout << "Enter base and exponent of first number ";
	cin >> b1 >> e1;
	cout << "Enter base and exponent of second number";
	cin >> b2 >> e2;
	prod(b1, e1, b2, e2);
	system("pause");
```

```
	return 0;
}
```



Practice Task 3

Create a structure course with some attributes i.e. course_ID, course title, credit etc.. Then Implement following 5 functions (Known as CRUDS operations which means CREATE, READ, UPDATE, DELETE, SEARCH operations):

Code:

```
#include<iostream>
#include<string.h>
using namespace std;
int total = 0;
struct stds {
	int course_id;
	string course_name;
	int crdt_hrs;
	string name;
	int roll_number;

};
void create(stds s[])
{
	int ch = 0;
	cout << "How to many syudents do you want to enter ";
	cin >> ch;
	if (total == 0)
	{
		total = ch + total;
		for (int i = 0; i < ch; i++)
		{
			cout << "\nEnter the Data of student " << i + 1 << endl << endl;
			cout << "Enter name \n";
			cin >> s[i].name;
			cout << endl;
			cout << "Enter course name  \n";
			cin >> s[i].course_name;
			cout << endl;
```

```
                        cout << "Enter course id\n";
                        cin >> s[i].course_id;
                        cout << endl;
                        cout << "Enter credit hours \n";
                        cin >> s[i].crdt_hrs;
                        cout << "Enter roll number of the student\n";
                        cin >> s[i].roll_number;
                        cout << endl;


                }
        }
        else
        {

                for (int i = total; i < ch + total; i++)
                {
                        cout << "\nEnter the Data of student " << i + 1 << endl << endl;
                        cout << "Enter name \n";
                        cin >> s[i].name;
                        cout << endl;
                        cout << "Enter course name  \n";
                        cin >> s[i].course_name;
                        cout << endl;
                        cout << "Enter course id\n";
                        cin >> s[i].course_id;
                        cout << endl;
                        cout << "Enter credit hours \n";
                        cin >> s[i].crdt_hrs;
                        cout << "Enter roll number of the student\n";
                        cin >> s[i].roll_number;
                }
                total = ch + total;
        }
}
void show(stds s[])
{
        if (total == 0)
        {
                cout << "No data is entered" << endl;
        }
        else {
                for (int i = 0; i < total; i++)
                {
                        cout << "\nData of Student " << i + 1 << endl << endl;
                        cout << "Name " << s[i].name << endl;
                        cout << "Roll no " << s[i].roll_number << endl;
                        cout << "Course name  " << s[i].course_name << endl;
                        cout << "Course id " << s[i].course_id << endl;
                        cout << "Credit hours " << s[i].crdt_hrs << endl;
                }
        }
}
void search(stds s[])
{
        if (total == 0)
        {
                cout << "No data is entered" << endl;
        }
        else {
                int rollno;
                cout << "Enter the roll no of student" << endl;
```

```
			cin >> rollno;
			for (int i = 0; i < total; i++)
			{
				if (rollno == s[i].roll_number)
				{
					cout << "Name " << s[i].name << endl;
					cout << "Roll no " << s[i].roll_number << endl;
					cout << "Course Name" << s[i].course_name << endl;
					cout << "Course id " << s[i].course_id << endl;
					cout << "Credit Hours " << s[i].crdt_hrs << endl;
				}
			}
		}
}
void update(stds s[])
{
		if (total == 0)
		{
			cout << "No data is entered" << endl;
		}
		else {
			int rollno;
			cout << "Enter the roll no of student which you want to update" << endl;
			cin >> rollno;
			for (int i = 0; i < total; i++)
			{
				if (rollno == s[i].roll_number)
				{
					cout << "\nPrevious data" << endl << endl;
					cout << "Data of Student " << i + 1 << endl;
					cout << "Name " << s[i].name << endl;
					cout << "Roll no " << s[i].roll_number << endl;
					cout << "Course Name " << s[i].course_name << endl;
					cout << "Course id  " << s[i].course_id << endl;
					cout << "Credit hours " << s[i].crdt_hrs << endl;
					cout << "\nEnter new data" << endl << endl;
					cout << "Enter name ";
					cin >> s[i].name;
					cout << "Enter Roll no ";
					cin >> s[i].roll_number;
					cout << "Enter course Name  ";
					cin >> s[i].course_name;
					cout << "Enter Course id  ";
					cin >> s[i].course_id;
					cout << "Enter credit hours ";
					cin >> s[i].crdt_hrs;
				}
			}
		}
}
void deleterecord(stds s[])
{
		if (total == 0)
		{
			cout << "No data is entered" << endl;
		}
		else {
			int a;
			cout << "Press 1 to delete all record" << endl;
			cout << "Press 2 to delete specific record" << endl;
			cin >> a;
			if (a == 1)
```

```
		{
			total = 0;
			cout << "All record is deleted..!!" << endl;
		}
		else if (a == 2)
		{
			int rollno;
			cout << "Enter the roll no of student which you wanted to delete"
<< endl;
			cin >> rollno;
			for (int i = 0; i < total; i++)
			{
				if (rollno == s[i].roll_number)
				{
					for (int j = i; j < total; j++)
					{
						s[j].course_id = s[j + 1].course_id;
						s[j].course_name = s[j + 1].course_name;
						s[j].crdt_hrs = s[j + 1].crdt_hrs;
						s[j].name = s[j + 1].name;
						s[j].roll_number = s[j + 1].roll_number;
					}
					total--;
					cout << "Your required record is deleted" << endl;
				}
			}
		}


		else
		{
			cout << "Invalid input";
		}
	}
}
int main()
{
	stds s[10];
	int value;
	while (true)
	{

		cout << "Press 1 to Enter data " << endl;
		cout << "Press 2 to Show data" << endl;
		cout << "Press 3 to Search data" << endl;
		cout << "Press 4 to Update data" << endl;
		cout << "Press 5 to Delete data" << endl;
		cout << "Press 6 to Exit" << endl;

		cin >> value;
		switch (value)
		{
		case 1:
			create(s);
			break;
		case 2:
			show(s);
			break;
		case 3:
			search(s);
			break;
		case 4:
```

```
                update(s);
                break;
        case 5:
                deleterecord(s);
                break;

        case 6:
                exit(0);
                break;
        default:
                cout << "Invalid input" << endl;
                break;
        }
        }
}
```