**NAME: Qazi M Aoun Farooqi**

**ROLL NO : 64**

**TASK 1:**

#include <iostream>

using namespace std;

void hcf(int a , int b){

while(b!=0){

int temp = b;

b=a%b;

a = temp;

}

cout<<"HCF of this number is :"<<a<<endl;

}

int main()

{

int a,b;

cout<<" Enter two Number :"<<endl;

cin>>a>>b;

hcf(a,b);

}

**Task 2 :**

**#include <iostream>**

**using namespace std;**

**int power(int base, int exponent) {**

**int result = 1;**

**for (int i = 0; i < exponent; ++i) {**

**result = result\*base;**

**}**

**cout<<result<<endl;**

**}**

**void computeProduct(int base1, int exp1, int base2, int exp2, int &product) {**

**if (base1 != base2) {**

**cout << "Bases are not the same. Cannot compute product." << endl;**

**product = -1;**

**cout<<product<<endl;**

**}**

**product = power(base1, exp1 + exp2);**

**}**

**int main() {**

**int base1, exp1, base2, exp2, result;**

**cout << "Enter base and exponent for first number: ";**

**cin >> base1 >> exp1;**

**cout << "Enter base and exponent for second number: ";**

**cin >> base2 >> exp2;**

**computeProduct(base1, exp1, base2, exp2, result);**

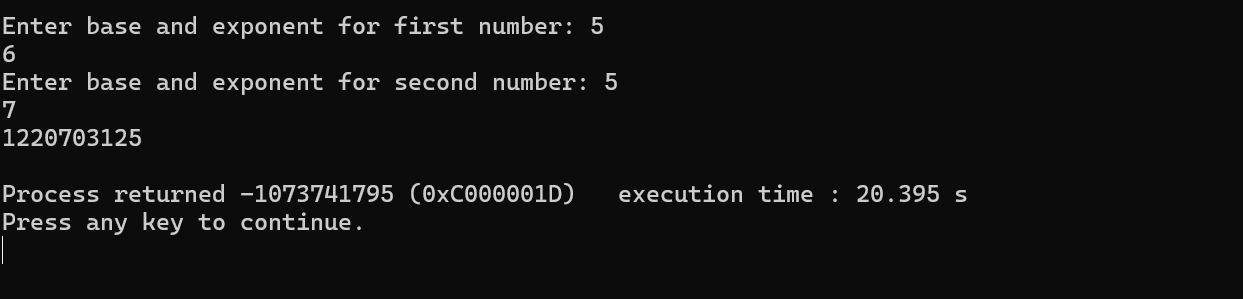
**if (result != -1) {**

**cout << "Product of power numbers: " << result << endl;**

**}**

**return 0;**

**}**



**TASK 3 :**

#include <iostream>

using namespace std;

const int SIZE = 5;

struct Course {

int course\_ID;

char course\_title[50];

int credit\_hrs;

bool isEmpty;

};

void addCourse(Course courses[], int index) {

cout << "Enter Course ID: ";

cin >> courses[index].course\_ID;

cout << "Enter Course Title (one word): ";

cin >> courses[index].course\_title;

cout << "Enter Credit Hours: ";

cin >> courses[index].credit\_hrs;

courses[index].isEmpty = false;

}

void updateCourse(Course courses[], int size) {

int id;

cout << "Enter Course ID to update: ";

cin >> id;

for (int i = 0; i < size; i++) {

if (!courses[i].isEmpty && courses[i].course\_ID == id) {

cout << "Enter new Course Title (one word): ";

cin >> courses[i].course\_title;

cout << "Enter new Credit Hours: ";

cin >> courses[i].credit\_hrs;

return;

}

}

cout << "Course not found.\n";

}

void deleteCourse(Course courses[], int size) {

int id;

cout << "Enter Course ID to delete: ";

cin >> id;

for (int i = 0; i < size; i++) {

if (!courses[i].isEmpty && courses[i].course\_ID == id) {

courses[i].isEmpty = true;

return;

}

}

cout << "Course not found.\n";

}

void searchCourse(Course courses[], int size) {

int id;

cout << "Enter Course ID to search: ";

cin >> id;

for (int i = 0; i < size; i++) {

if (!courses[i].isEmpty && courses[i].course\_ID == id) {

cout << "Course ID: " << courses[i].course\_ID << "\n";

cout << "Course Title: " << courses[i].course\_title << "\n";

cout << "Credit Hours: " << courses[i].credit\_hrs << "\n";

return;

}

}

cout << "Course not found.\n";

}

void displayCourses(Course courses[], int size) {

for (int i = 0; i < size; i++) {

if (!courses[i].isEmpty) {

cout << "Course ID: " << courses[i].course\_ID << "\n";

cout << "Course Title: " << courses[i].course\_title << "\n";

cout << "Credit Hours: " << courses[i].credit\_hrs << "\n\n";

}

}

}

int main() {

Course courses[SIZE];

for (int i = 0; i < SIZE; i++) {

courses[i].isEmpty = true;

}

int choice;

do {

cout << "\n1. Add Course\n2. Update Course\n3. Delete Course\n4. Search Course\n5. Display All Courses\n6. Exit\n";

cout << "Enter choice: ";

cin >> choice;

switch (choice) {

case 1:

for (int i = 0; i < SIZE; i++) {

if (courses[i].isEmpty) {

addCourse(courses, i);

break;

}

}

break;

case 2: updateCourse(courses, SIZE); break;

case 3: deleteCourse(courses, SIZE); break;

case 4: searchCourse(courses, SIZE); break;

case 5: displayCourses(courses, SIZE); break;

case 6: cout << "Exiting...\n"; break;

default: cout << "Invalid choice.\n";

}

} while (choice != 6);

return 0;

}

