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

int subtract() {
    int a, b;
    cout << "Enter two numbers to subtract: ";
    cin >> a >> b;
    cout << "The difference is: " << a - b << endl;
    return 0;
}

int Multiply() {
    int a, b;
    cout << "Enter two numbers to multiply: ";
    cin >> a >> b;
    cout << "The product is: " << a * b << endl;
    return 0;
}

int Divide() {
    int a, b;
    cout << "Enter two numbers to divide: ";
    cin >> a >> b;
    if (b != 0) {
        cout << "The quotient is: " << a / b << endl;
    } else {
        cout << "Error: Division by zero is not allowed." << endl;
    }
    return 0;
}

int sum() {
    int a, b;
    cout << "Enter two numbers to add: ";
    cin >> a >> b;
    cout << "The sum is: " << a + b << endl;
    return 0;
}
```

```cpp
int menu() {
    int n;
    cout << "Menu Options:" << endl;
    cout << "1. Add 2 Numbers" << endl;
    cout << "2. Subtract 2 Numbers" << endl;
    cout << "3. Multiply 2 Numbers" << endl;
    cout << "4. Divide 2 Numbers" << endl;
    cout << "5. Exit" << endl;
    cout << "Please select an option (1-5): ";
    cin >> n;
    return n;
}

void menuOption() {
    while (true) {
        int option = menu();

        if (option == 1) {
            sum();
        }
        else if (option == 2) {
            subtract();
        }
        else if (option == 3) {
            Multiply();
        }
        else if (option == 4) {
            Divide();
        }
        else if (option == 5) {
            cout << "Exiting the program." << endl;
            break;
        }
        else {
            cout << "Error message. Please select a valid option (1-5)." << endl;
        }
    }
```

```
void menuOption() {
    while (true) {
        int option = menu();

        if (option == 1) {
            sum();
        }
        else if (option == 2) {
            subtract();
        }
        else if (option == 3) {
            Multiply();
        }
        else if (option == 4) {
            Divide();
        }
        else if (option == 5) {
            cout << "Exiting the program." << endl;
            break;
        }
        else {
            cout << "Error message. Please select a valid option (1-5)." << endl;
        }
    }
}

int main() {
    menuOption();
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int max3(int a, int b, int c){
    int m = a;
    if (b > m) {
        m = b;}
    if (c > m) {
        m = c;}
    return m;}
int max4(int a, int b, int c, int d){
    int m =max3(a,b,c);
    if (m > d) {
        return m;
    } else {
        return d;
    }}
int max5(int a, int b, int c, int d, int e){
    int m =max4(a,b,c,d);
    if (m > e) {
        return m;
    } else {
        return e;
    }}
int max6(int a, int b, int c, int d, int e, int f){
    int m =max5(a,b,c,d,e);
    if (m > f) {
        return m;
    } else {
        return f;
    }}
int main() {
    int a,b,c,d,e,f;
    cout << "Enter six numbers: ";
    cin >> a >> b >> c >> d >> e >> f;
    int res =  max6(a,b,c,d,e,f);
    cout << "The maximum number is: " << res << endl;
    return 0;
}
```

```cpp
#include <iostream>
#include <string>
using namespace std;
 string rev_string(const string &str ) {
    string res = "";
    for (int i = str.length() - 1; i >= 0 ; i--) {
        res += str[i];
    }
    return res;
}

int main() {
    string input;
    cout << "Enter a string: ";
    cin >> input;
    string reversed = rev_string(input);
    cout << "Reversed string: " << reversed << endl;
    return 0;
}

```

```cpp
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
bool ispalindrome(int arr[], int n) {
    for (int i = 0; i < n / 2; i++) {
        if (arr[i] != arr[n - i - 1]) {
            return false;
        }
    }
    return true;
}
int main() {
    int arr[1000];
    int n;
    cout << "Enter the number of elements in the array: ";
    cin >> n;
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    if (ispalindrome(arr, n)) {
        cout << "The array is a palindrome." << endl;
    } else {
        cout << "The array is not a palindrome." << endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
#include <string>
#include <algorithm>
#include <cmath>
using namespace std;
void set_powers(int arr[], int len , int m ) {
    int n = 0;
    for (int i = 0; i <= len; i++) {
        arr[i] = pow(m, i);
    }
   }
int main() {
    int len, m;
    cout << "Enter the Number: ";
    cin >> len;
    cout << "Enter the Power: ";
    cin >> m;
    int arr[len];
    set_powers(arr, len, m);
    cout << "The array Is: ";
    for (int i = 0; i < len; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}

```

```cpp
#include <iostream>
#include <string>
#include <algorithm>
#include <cmath>
using namespace std;
bool is_prime(int num) {
    if (num <= 1) return false;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0)
            return false;
    }
    return true;
}
int nth_prime(int n) {
    int count = 0;
    int num = 2;
    while (true) {
        if (is_prime(num)) {
            count++;
            if (count == n) {
                return num;
            }
        }
        num++;
    }
}

int main() {
    int n;
    cout << "Enter the position of the prime number: ";
    cin >> n;
    int prime = nth_prime(n);
    cout << "The " << n << "-th prime number is: " << prime << endl;

    return 0;
}
```

```cpp
#include <iostream>
#include <string>
#include <algorithm>
#include <cmath>
using namespace std;
bool starts_with(string input, string pattern, int pos) {
    for (int i = 0; i < pattern.length(); i++) {
        if (pos + i >= input.length() || input[pos + i] != pattern[i]) {
            return false;
        }}
    return true;}
string replace_str(string input, string pattern, string to) {
    string result = "";
    int i = 0;
    while (i < input.length()) {
        if (starts_with(input, pattern, i)) {
            result += to;
            i += pattern.length();
        } else {
            result += input[i];
            i++;
        }}
    return result;}
int main() {
    string input, pattern, to;
    cout << "Enter the input string: ";
    cin >> input;
    cout << "Enter the pattern to replace: ";
    cin >> pattern;
    cout << "Enter the replacement string: ";
    cin >> to;
    string output = replace_str(input, pattern, to);
    cout << "Result: " << output << endl;
    return 0;
}
```

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

const int max_specialization = 20;
const int max_queue = 5;

struct Patient {
    string name;
    int status; };

Patient hospital[max_specialization + 1][max_queue];
int count[max_specialization + 1] = {0};

void Add() {
    int spec, status;
    string name;

    cout << "Enter specialization (1-20): ";
    cin >> spec;

    if (spec < 1 || spec > 20) {
        cout << "Invalid specialization.\n";
        return;
    }

    if (count[spec] >= max_queue) {
        cout << "Sorry, no available slots in this specialization.\n";
        return;
    }

    cout << "Enter patient name: ";
    cin >> name;
    cout << "Enter status (0 = regular, 1 = urgent): ";
    cin >> status;

    if (status == 1) {
```

```
    if (status == 1) {
        for (int i = count[spec]; i > 0; i--) {
            hospital[spec][i] = hospital[spec][i - 1];
        }
        hospital[spec][0] = {name, status};
    } else {
        hospital[spec][count[spec]] = {name, status};
    }

    count[spec]++;
    cout << "Patient added successfully.\n";
}
void Print() {
    for (int spec = 1; spec <= max_specialization; spec++) {
        if (count[spec] > 0) {
            cout << "Specialization " << spec << ":\n";
            for (int i = 0; i < count[spec]; i++) {
                cout << "  [Spec " << spec << "] " << hospital[spec][i].name;
                if (hospital[spec][i].status == 1)
                    cout << " (urgent)";
                cout << "\n";
            }
        }
    }
}

void Get() {
    int spec;
    cout << "Enter specialization (1-20): ";
    cin >> spec;

    if (spec < 1 || spec > 20 || count[spec] == 0) {
        cout << "No patients in this specialization.\n";
        return;
    }
```

```
void Get() {
    int spec;
    cout << "Enter specialization (1-20): ";
    cin >> spec;

    if (spec < 1 || spec > 20 || count[spec] == 0) {
        cout << "No patients in this specialization.\n";
        return;
    }

    cout << hospital[spec][0].name << ", please go with the doctor.\n";

    for (int i = 1; i < count[spec]; i++) {
        hospital[spec][i - 1] = hospital[spec][i];
    }

    count[spec]--;
}
void Choice() {
    int choice;
    while (true) {
        cout << "\nEnter your choice:\n";
        cout << "1. Add new patient\n";
        cout << "2. Print all patients\n";
        cout << "3. Get new patient\n";
        cout << "4. Exit\n";
        cout << "Your choice: ";
        cin >> choice;
        if (choice == 1){
            Add();
        } else if (choice == 2){
            Print();
        } else if (choice == 3){
            Get();
        } else if (choice == 4){
            cout << "Exiting the program.\n";
            break;
```

```
    count[spec]--;
}
void Choice() {
    int choice;
    while (true) {
        cout << "\nEnter your choice:\n";
        cout << "1. Add new patient\n";
        cout << "2. Print all patients\n";
        cout << "3. Get new patient\n";
        cout << "4. Exit\n";
        cout << "Your choice: ";
        cin >> choice;
        if (choice == 1){
            Add();
        } else if (choice == 2){
            Print();
        } else if (choice == 3){
            Get();
        } else if (choice == 4){
            cout << "Exiting the program.\n";
            break;
        } else {
            cout << "Invalid choice. Please try again.\n";
        }
    }
}
int main() {
    Choice();
    return 0;
}
```