LAB 2

NAME PARI BATRA

SECTION BCS3A

ROLL NO 24K3115

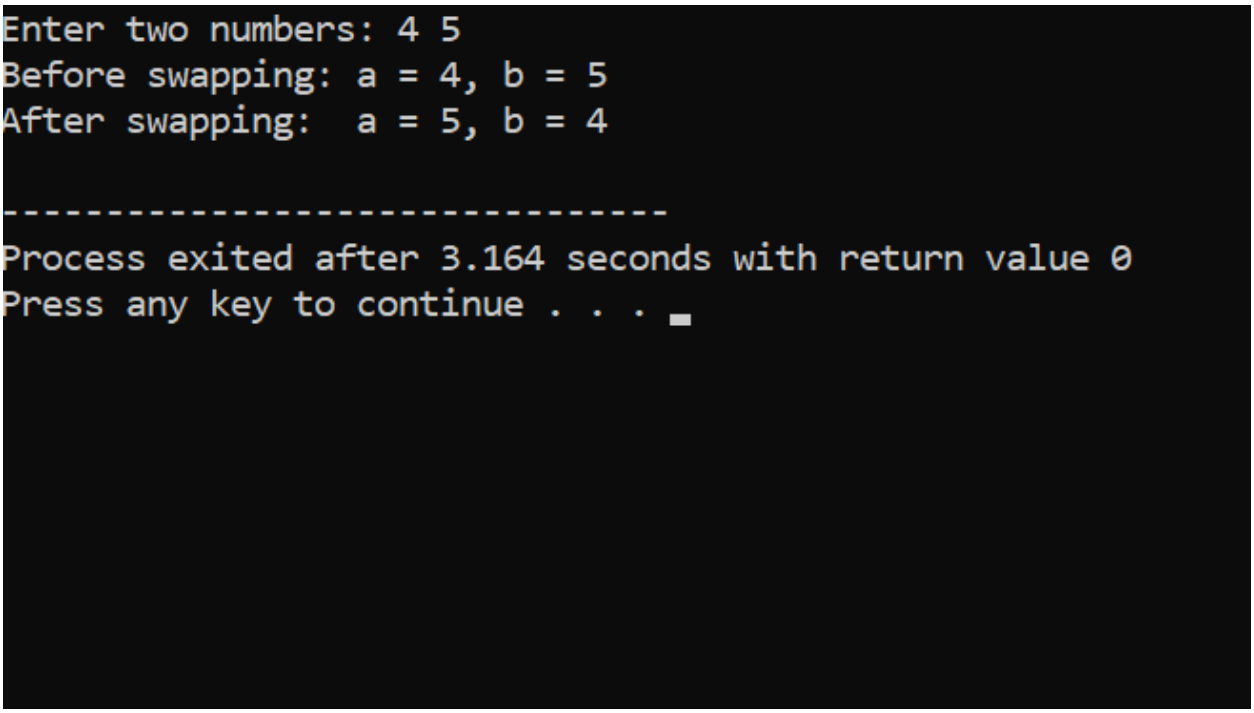
TASK # 01

Write a program in C++ that creates a function named SwapValues which takes two pointers as

arguments and swaps their values without using a third variable.

lab 2 task 1.cpp

```cpp
//task no 1 by pari batra 24k3115
#include<iostream>

using namespace std;

// Function to swap two values
void SwapValues(int* a, int* b) {
    if (a == b) return;

    *a = *a + *b;   // Step 1: add values
    *b = *a - *b;   // Step 2: subtract to get original *a
    *a = *a - *b;   // Step 3: subtract again to get original *b
}

int main() {
    int a, b;

    cout<< "Enter two numbers: ";
    cin >> a >> b;

    cout << "Before swapping: a = " << a << ", b = " << b<< endl;

    SwapValues(&a, &b);

    cout << "After swapping:  a = " << a << ", b = " << b << endl;

    return 0;
}
```



TASK # 02

Write a program in C++ that creates a function named FirstAndLastIndex which takes a string, a

character, and two pointer variables as arguments. It should calculate and return the first and last

occurrence of the character in the string using pointers.

lab 2 task 1.cpp | [*] lab2 task 2.cpp

```cpp
//Lab2 task2 by pari batra 24k3115
#include<iostream>
#include <string>
using namespace std;


void firstandlastindex(const string &str, char ch, int* first, int* last) {
    *first = -1;  // default value if not found
    *last = -1;

    const char* ptr = str.c_str(); // pointer to string data
    for (int i = 0; *(ptr + i) != '\0'; i++) {
        if (*(ptr + i) == ch) {
            if (*first == -1) {
                *first = i;   //  first occurrence
            }
            *last = i;        //  last occurrence
        }
    }
}

int main() {
    string str;
    char ch;
    int firstindex, lastindex;

    cout << "Enter a string: ";
    getline(cin, str);

    cout << "Enter the  character that you want to search: ";
    cin >> ch;

    firstandlastindex(str, ch, &firstindex, &lastindex);

    if (firstindex == -1) {
        cout << "Character '" << ch << "' not found." << endl;
    } else {
        cout << "First occurrence of '" << ch << "' is at index: " << firstindex << endl;
        cout << "Last occurrence of '" << ch << "' is at index: " << lastindex << endl;
```

```
Enter a string: materialistic
Enter the  character that you want to search: i
First occurrence of 'i' is at index: 5
Last occurrence of 'i' is at index: 11

--------------------------------
Process exited after 11.34 seconds with return value 0
Press any key to continue . . .
```

TASK # 03

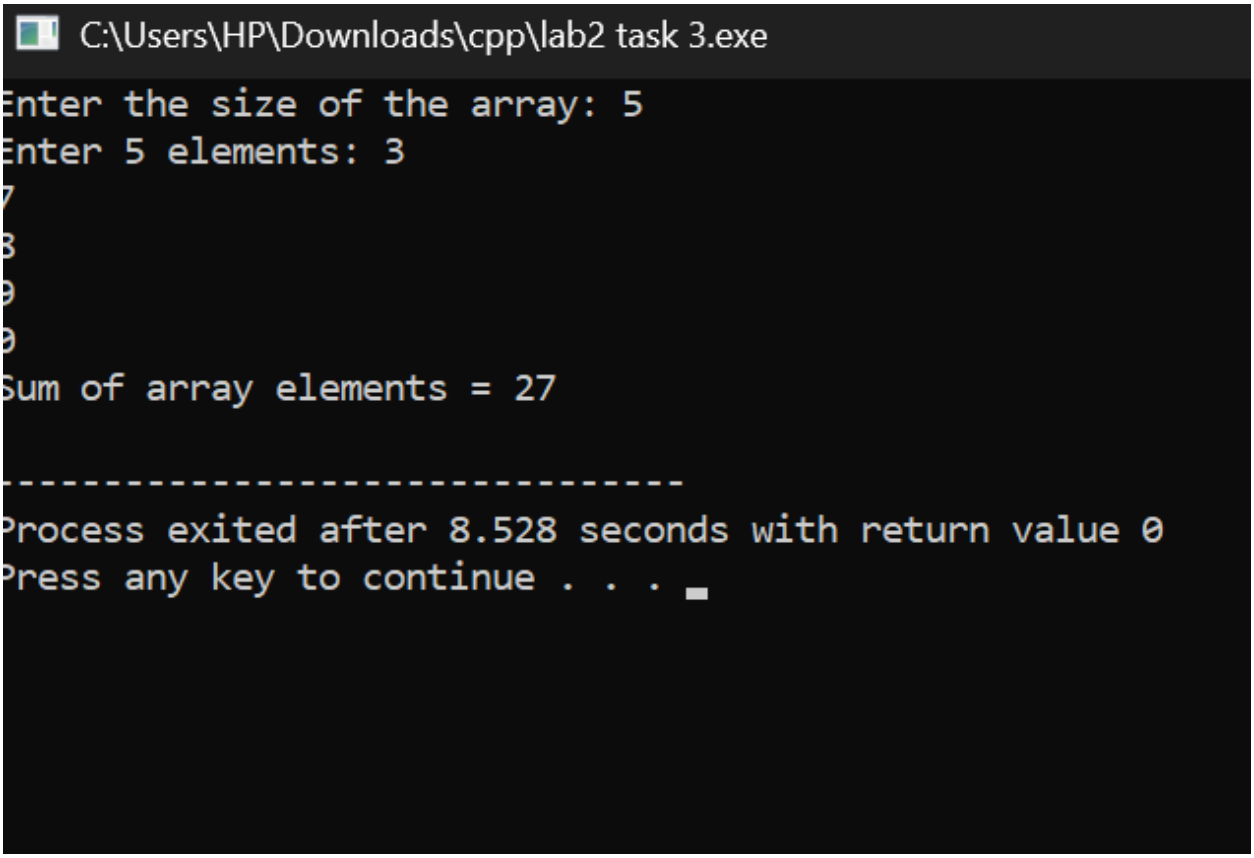
Write a program in C++ that creates a function named sumArray which takes an array and its size

as arguments (using a pointer) and calculates the sum of all the elements in the array. The

function should use pointer arithmetic to access the elements.

lab 2 task 1.cpp | [*] lab2 task 2.cpp | lab2 task 3.cpp

```cpp
//task3 lab2 by pari batra 24k3115

#include<iostream>
using namespace std;

// Function to calculate sum of array using pointer arithmetic
int sumArray(int* arr, int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += *(arr + i);  // accessing elements via pointer arithmetic
    }
    return sum;
}

int main() {
    int size;

    cout << "Enter the size of the array: ";
    cin >> size;

    int* arr = new int[size]; // dynamically allocate array

    cout << "Enter " << size << " elements: ";
    for (int i = 0; i < size; i++) {
        cin >> *(arr + i);  // input using pointer arithmetic
    }

    int result = sumArray(arr, size);

    cout << "Sum of array elements = " << result << endl;

    delete[] arr; // free memory
    return 0;
}

```

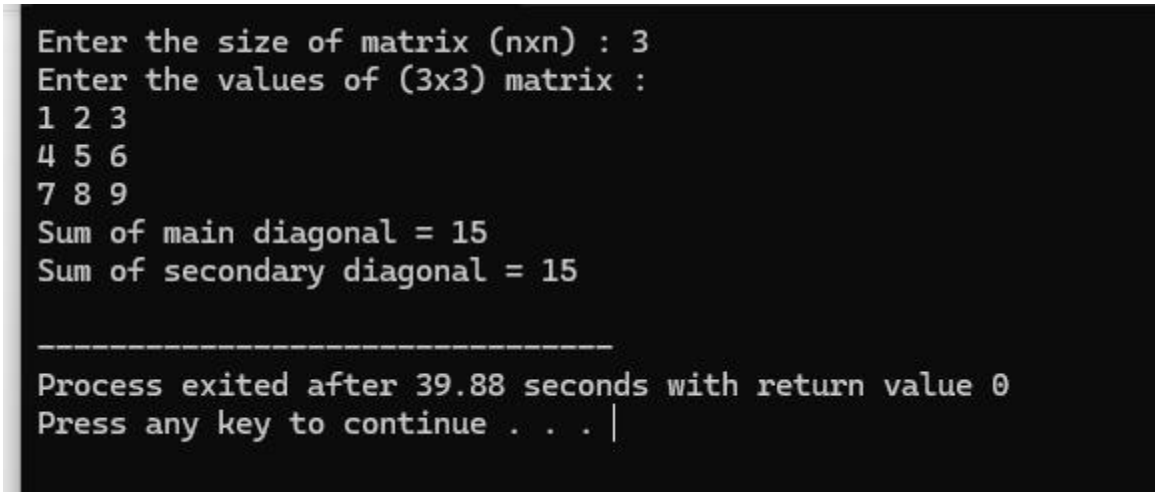


TASK # 04

Write a program in C++ that dynamically allocates memory for a square matrix (NxN), takes input from the user, and calculates the sum of both the main diagonal and the secondary diagonal. The program should then display both sums and the matrix

```cpp
//task no 4 by pari batra 24k3115
#include <iostream>
using namespace std;

// Function to calculate sum of main diagonal elements
int sumOfMainDiagonal(int **matrix, int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += matrix[i][i];
    }
    return sum;
}


// Function to calculate sum of secondary diagonal elements
int sumOfSecondaryDiagonal(int **matrix, int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += matrix[i][n - i - 1];
    }
    return sum;
}

int main() {
    int n;
    cout << "Enter the size of matrix (nxn) : ";
    cin >> n;

    // Memory allocation
    int *matrix = new int[n];
    for (int i = 0; i < n; i++) {
        matrix[i] = new int[n];
    }

    cout << "Enter the values of (" << n << "x" << n << ") matrix : " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> matrix[i][j];
        }
```

```cpp

int main() {
    int n;
    cout << "Enter the size of matrix (nxn) : ";
    cin >> n;

    // Memory allocation
    int *matrix = new int[n];
    for (int i = 0; i < n; i++) {
        matrix[i] = new int[n];
    }

    cout << "Enter the values of (" << n << "x" << n << ") matrix : " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> matrix[i][j];
        }
    }

    \
    int mainSum = sumOfMainDiagonal(matrix, n);
    int secSum = sumOfSecondaryDiagonal(matrix, n);

    cout << "Sum of main diagonal = " << mainSum << endl;
    cout << "Sum of secondary diagonal = " << secSum << endl;

    // Free memory
    for (int i = 0; i < n; i++) {
        delete[] matrix[i];
    }
    delete[] matrix;

    return 0;
}
```



TASK # 05

Write a program in C++ that dynamically allocates memory for two strings, takes input for both

strings from the user, and concatenates them into a third string. The program should display the

original strings and the concatenated result.

```cpp
//task no 5 vy pari batra 24k3115
#include<iostream>
#include <cstring>   // for strlen, strcpy, strcat
using namespace std;

int main() {
    // Dynamically allocate memory for two strings
    char *str1 = new char[100];
    char *str2 = new char[100];

    cout << "Enter first string: ";
    cin.getline(str1, 100);

    cout << "Enter second string: ";
    cin.getline(str2, 100);

    // Allocate memory for concatenated string
    int totalLength = strlen(str1) + strlen(str2) + 1; // +1 for '\0'
    char *concatStr = new char[totalLength];

    // Copy and concatenate
    strcpy(concatStr, str1);
    strcat(concatStr, str2);

    // Display results
    cout << "\nFirst String: " << str1;
    cout << "\nSecond String: " << str2;
    cout << "\nConcatenated String: " << concatStr << endl;

    // Free
    delete[] str1;
    delete[] str2;
    delete[] concatStr;

    return 0;
}
```

```
Enter first string: cat
Enter second string: dog

First String: cat
Second String: dog
Concatenated String: catdog

--------------------------------
Process exited after 8.718 seconds with return value 0
Press any key to continue . . .
```