Assignment 2

Name -Shreyas Anil Dimbar.

Class – S.Y. CSE

Batch – S6

Q1 ) Create a memory for int N, char C and float F variables at run time using malloc () function and before exiting the program release the memory allocated at run time by using a "free" function. After reading the values, display those on a console.

#include <stdio.h>

#include <stdlib.h>

int main() {

    int \*N = (int \*)malloc(sizeof(int));

    char \*C = (char \*)malloc(sizeof(char));

    float \*F = (float \*)malloc(sizeof(float));

    printf("Enter integer n : ");

    scanf("%d", N);

    printf("Enter character c : ");

    scanf(" %c", C);  // space before %c to consume any leftover newline

    printf("Enter float number : ");

    scanf("%f", F);

    printf("Integer : %d\n", \*N);

    printf("Character : %c\n", \*C);

    printf("Float : %.2f\n", \*F);

    free(N);

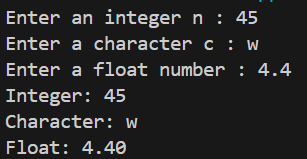
    free(C);

    free(F);

    return 0;

}

Output:



Q2) Given an input N and array A. Find the largest element using dynamic memory allocation function "calloc".

#include <stdio.h>

#include <stdlib.h>

int main() {

    int N, max;

    printf("Enter number of elements (N): ");

    scanf("%d", &N);

    int \*A = (int \*)calloc(N, sizeof(int));

    printf("Enter %d elements:\n", N);

    for (int i = 0; i < N; i++) {

        scanf("%d", &A[i]);

    }

    max = A[0];

    for (int i = 1; i < N; i++) {

        if (A[i] > max) {

            max = A[i];

        }

    }

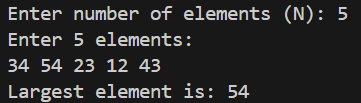
    printf("Largest element is: %d\n", max);

    free(A);

    return 0;

}

Output:



Q3) Given an array of integers, allocate memory dynamically for array and find the sum of N even numbers.

#include <stdio.h>

#include <stdlib.h>

int main() {

    int N,sum = 0;

    printf("Enter number of elements (N): ");

    scanf("%d", &N);

    int \*arr = (int \*)malloc(N \* sizeof(int));

    printf("Enter %d integers:\n", N);

    for (int i = 0; i < N; i++) {

        scanf("%d", &arr[i]);

    }

    for (int i = 0; i < N; i++) {

        if (arr[i] % 2 == 0) {

            sum += arr[i];

        }

    }

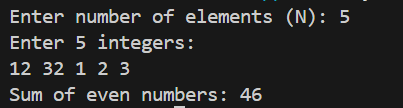
    printf("Sum of even numbers: %d\n", sum);

    free(arr);

    return 0;

}

Output:



Q 4) Given two integer arrays, swap the two arrays using pointers.

#include <stdio.h>

void swapArrays(int \*a, int \*b, int n) {

    for (int i = 0; i < n; i++) {

        int temp = \*(a + i);

        \*(a + i) = \*(b + i);

        \*(b + i) = temp;

    }

}

int main() {

    int n = 5;

    int arr1[5] = {1, 2, 3, 4, 5};

    int arr2[5] = {6, 7, 8, 9, 10};

    swapArrays(arr1, arr2, n);

    printf("Array 1: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr1[i]);

    }

    printf("\nArray 2: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr2[i]);

    }

    return 0;

}

Output:



Q5) Write a program using function that accepts an integer N. The function should return binary of N using recursion.

#include <stdio.h>

void printBinary(int N) {

    if (N > 0) {

        printBinary(N / 2);

        printf("%d", N % 2);

    }

}

int main() {

    int N;

    printf("Enter an integer: ");

    scanf("%d", &N);

    if (N == 0) {

        printf("Binary: 0\n");

    }

    else {

        printf("Binary: ");

        printBinary(N);

        printf("\n");

    }

    return 0;

}

Output:



Q6) Write a program to implement Tower of Hanoi problem using recursion.

#include <iostream>

using namespace std;

void towerOfHanoi(int n, char source, char auxiliary, char destination) {

    if (n == 1) {

        cout << source << " to " << destination << endl;

        return;

    }

    towerOfHanoi(n - 1, source, destination, auxiliary);

    cout<< source << " to " << destination << endl;

    towerOfHanoi(n - 1, auxiliary, source, destination);

}

int main() {

    int n;

    cout << "Enter number of disks: ";

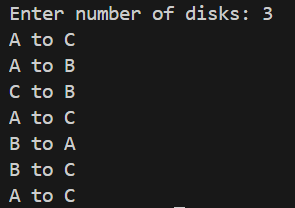
    cin >> n;

    towerOfHanoi(n, 'A', 'B', 'C');

    return 0;

}

Output:



Q7) Write a Program to Find the Biggest Number in a one-dimensional unsorted array of Numbers using recursive technique.

#include <iostream>

using namespace std;

int findMax(int arr[], int n) {

    if (n == 1){

        return arr[0];

    }

    return max( arr[n - 1] , findMax(arr, n - 1) );

}

int main() {

    int n;

    cout << "Enter number of elements: ";

    cin >> n;

    int arr[n];

    cout << "Enter elements:"<<endl;

    for (int i = 0; i < n; ++i)

        cin >> arr[i];

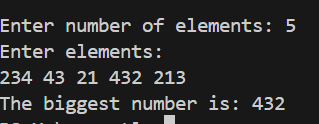
    int maxElement = findMax(arr, n);

    cout << "The biggest number is: " << maxElement << endl;

    return 0;

}

Output:



Q 8) Write a program to merge two files and generate a third file. Merge of the two files should happen for alternative lines. i.e. 1st line from 1st file, 2nd line from 2nd file, 3rd line from 1st file and so on.

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int main(){

    ifstream file1("file1.txt");

    ifstream file2("file2.txt");

    ofstream merge("merge.txt");

    bool check = true;

    string s1;

    string s2;

    while(check){

        check = false;

        if(getline(file1,s1)){

            merge << s1 << endl;

            check = true ;

        }

        if(getline(file2,s2)){

            merge << s2 << endl;

            check = true ;

        }

    }

    file1.close();

    file2.close();

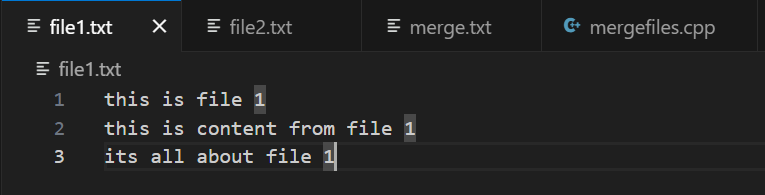
    merge.close();

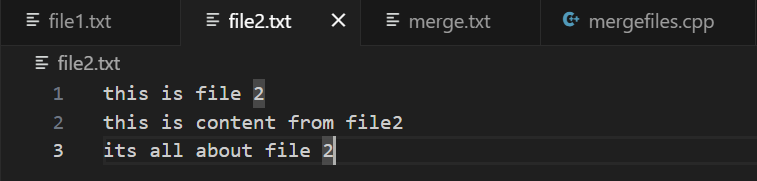
    cout << "files merged into merge.txt" << endl;

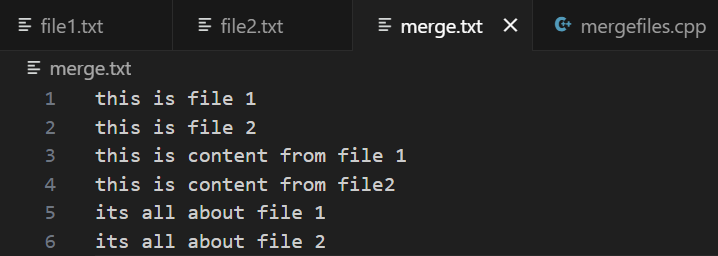
    return 0;

}

Output:







Q9) Write a program to count occurrences of all words in a file.

#include <iostream>

#include <string>

#include <fstream>

#include <map>

using namespace std;

int main(){

    ifstream file("words.txt");

    string word;

    map<string,int>occ;

    while( file >> word ){

        occ[word]++;

    }

    file.close();

    cout << "word occurances:"<<endl;

    for( const auto &val : occ ){

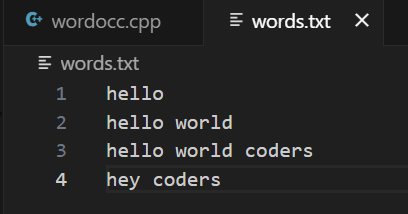
        cout << val.first << " : " << val.second << endl ;

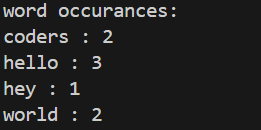
    }

    return 0;

}

Output:





Q10) Write a program to find sum of all the command line arguments.

#include<iostream>

#include<cstdlib>

using namespace std;

 int main(int argc, char \*argv[])

{

  int sum=0;

  for(int j=1;j<argc;j++){

    sum+=atoi(argv[j]);

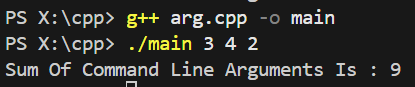
  }

  cout<<"Sum Of Command Line Arguments Is : "<<sum<<endl;

  return 0;

}

Output:



Q11 ) Write a program using indirect recursion to print numbers from n to 1. Use two functions: void printEven (int n) – which prints the number if it's even and calls printOdd (n - 1) void printOdd (int n) – which prints the number if it's odd and calls printEven (n - 1) The recursion should continue until n becomes 0.

#include <iostream>

using namespace std;

void printOdd(int n);

void printEven(int n) {

    if (n == 0)

        return;

    if (n % 2 == 0)

        cout << n << " ";

    printOdd(n - 1);

}

void printOdd(int n) {

    if (n == 0)

        return;

    if (n % 2 != 0)

        cout << n << " ";

    printEven(n - 1);

}

int main() {

    int n;

    cout << "Enter a number: ";

    cin >> n;

    if(n%2 == 0)

        printEven(n);

    else

        printOdd(n);

    cout << endl;

    return 0;

}

Output:

