**ASSIGNMENT-1**

**QUESTION-1)**

**INPUT**

**#include <stdio.h>**

**#include <math.h>**

**int isArmstrong(int num) {**

**int sum=0, temp, remainder, digits=0;**

**temp=num;**

**while (temp != 0)**

**{**

**digits++;**

**temp /= 10;**

**}**

**temp=num;**

**while (temp != 0)**

**{**

**remainder = temp % 10;**

**sum += pow(remainder, digits);**

**temp /= 10;**

**}**

**if (sum == num){**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**int main() {**

**int num;**

**printf("Enter a number: ");**

**scanf("%d", &num);**

**if (isArmstrong(num)) {**

**printf("%d is an Armstrong number.\n", num);**

**} else {**

**printf("%d is not an Armstrong number.\n", num);**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/P44QOy69YA.o**

**Enter a number: 153**

**153 is an Armstrong number.**

**=== Code Execution Successful ===**

**QUESTION-2)**

**INPUT**

**#include <stdio.h>**

**int calculateHCF(int a, int b) {**

**int min = (a < b) ? a : b;**

**for (int i = min; i >= 1; i--) {**

**if (a % i == 0 && b % i == 0) {**

**return i;**

**}**

**}**

**return 1;**

**}**

**int main() {**

**int num1, num2;**

**printf("Enter two integers: ");**

**scanf("%d %d", &num1, &num2);**

**int hcf = calculateHCF(num1, num2);**

**printf("The HCF of %d and %d is: %d\n", num1, num2, hcf);**

**return 0;**

**}**

**Output**

**/tmp/3A3HC5p6SF.o**

**Enter two integers: 2 42**

**The HCF of 2 and 42 is: 2**

**=== Code Execution Successful ===**

**QUESTION-3)**

**INPUT**

**#include <stdio.h>**

**int subtract(int a, int b) {**

**while (b != 0) {**

**int borrow = (~a) & b;**

**a = a ^ b;**

**b = borrow << 1;**

**}**

**return a;**

**}**

**int main() {**

**int num1, num2;**

**printf("Enter two integers: ");**

**scanf("%d %d", &num1, &num2);**

**printf("The result of %d - %d is: %d\n", num1, num2, subtract(num1, num2));**

**return 0;**

**}**

**Output**

**/tmp/BPlUf084mW.o**

**Enter two integers: 55 4**

**The result of 55 - 4 is: 51**

**=== Code Execution Successful ===**

**QUESTION-4)**

**INPUT**

**#include <stdio.h>**

**//METHOD-1**

**void swapUsingTemp(int \*a, int \*b) {**

**int temp = \*a;**

**\*a = \*b;**

**\*b = temp;**

**}**

**//METHOD-2**

**void swapUsingArithmetic(int \*a, int \*b) {**

**\*a = \*a + \*b;**

**\*b = \*a - \*b;**

**\*a = \*a - \*b;**

**}**

**//METHOD-3**

**void swapUsingXOR(int \*a, int \*b) {**

**if (a != b) {**

**\*a = \*a ^ \*b;**

**\*b = \*a ^ \*b;**

**\*a = \*a ^ \*b;**

**}**

**}**

**//METHOD-4**

**void swapUsingPointer(int \*\*a, int \*\*b) {**

**int \*temp = \*a;**

**\*a = \*b;**

**\*b = temp;**

**}**

**int main() {**

**int num1, num2;**

**printf("Enter two integers: ");**

**scanf("%d %d", &num1, &num2);**

**printf("\nOriginal values: num1 = %d, num2 = %d\n", num1, num2);**

**//METHOD-1**

**swapUsingTemp(&num1, &num2);**

**printf("\nAfter swapping using temp variable: num1 = %d, num2 = %d\n", num1, num2);**

**swapUsingTemp(&num1, &num2);**

**//METHOD-2**

**swapUsingArithmetic(&num1, &num2);**

**printf("\nAfter swapping using arithmetic: num1 = %d, num2 = %d\n", num1, num2);**

**swapUsingArithmetic(&num1, &num2);**

**//METHOD-3**

**swapUsingXOR(&num1, &num2);**

**printf("\nAfter swapping using XOR: num1 = %d, num2 = %d\n", num1, num2);**

**swapUsingXOR(&num1, &num2);**

**//METHOD-4**

**int \*ptr1 = &num1;**

**int \*ptr2 = &num2;**

**swapUsingPointer(&ptr1, &ptr2);**

**printf("\nAfter swapping using Pointer manipulation: num1 = %d, num2 = %d\n", num1, num2);**

**return 0;**

**}**

**OUTPUT**

**/tmp/fmlA7dbc7A.o**

**Enter two integers: 4 5**

**Original values: num1 = 4, num2 = 5**

**After swapping using temp variable: num1 = 5, num2 = 4**

**After swapping using arithmetic: num1 = 5, num2 = 4**

**After swapping using XOR: num1 = 5, num2 = 4**

**After swapping using Pointer manipulation: num1 = 4, num2 = 5**

**=== Code Execution Successful ===**

**QUESTION-5)**

**INPUT**

**#include <stdio.h>**

**int is\_perfect\_number(int num) {**

**int sum\_of\_divisors = 0;**

**for (int i = 1; i < num; i++) {**

**if (num % i == 0) {**

**sum\_of\_divisors += i;**

**}**

**}**

**if (sum\_of\_divisors == num) {**

**return 1;**

**} else {**

**return 0;**

**}**

**}**

**int main() {**

**int number;**

**printf("Enter a number: ");**

**scanf("%d", &number);**

**if (is\_perfect\_number(number)) {**

**printf("%d is a Perfect Number.\n", number);**

**} else {**

**printf("%d is not a Perfect Number.\n", number);**

**}**

**return 0;**

**}**

**Output**

**/tmp/q54vhjGE9I.o**

**Enter a number: 6**

**6 is a Perfect Number.**

**=== Code Execution Successful ===**

**/tmp/geC1CuNFZL.o**

**Enter a number: 23**

**23 is not a Perfect Number.**

**=== Code Execution Successful ===**

**QUESTION-6)**

**INPUT**

**#include <stdio.h>**

**int main() {**

**int x, y;**

**printf("Enter the x and y coordinates: ");**

**scanf("%d %d", &x, &y);**

**if (x > 0 && y > 0)**

**{**

**printf("The coordinate point (%d, %d) lies in the First quadrant.\n", x, y);**

**}**

**else if (x < 0 && y > 0)**

**{**

**printf("The coordinate point (%d, %d) lies in the Second quadrant.\n", x, y);**

**}**

**else if (x < 0 && y < 0)**

**{**

**printf("The coordinate point (%d, %d) lies in the Third quadrant.\n", x, y);**

**}**

**else if (x > 0 && y < 0)**

**{**

**printf("The coordinate point (%d, %d) lies in the Fourth quadrant.\n", x, y);**

**}**

**else if (x == 0 && y != 0)**

**{**

**printf("The coordinate point (%d, %d) lies on the Y-axis.\n", x, y);**

**}**

**else if (y == 0 && x != 0)**

**{**

**printf("The coordinate point (%d, %d) lies on the X-axis.\n", x, y);**

**}**

**else {**

**printf("The coordinate point (%d, %d) lies at the Origin.\n", x, y);**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/gAIwVxWffY.o**

**Enter the x and y coordinates: 8 9**

**The coordinate point (8, 9) lies in the First quadrant.**

**=== Code Execution Successful ===**

**QUESTION-7)**

**INPUT**

**#include <stdio.h>**

**#include <math.h>**

**int binary\_to\_decimal(int binary) {**

**int decimal = 0, remainder, i = 0;**

**while (binary != 0) {**

**remainder = binary % 10;**

**decimal+=remainder \* pow(2, i);**

**i++;**

**binary /= 10;**

**}**

**return decimal;**

**}**

**void decimal\_to\_binary(int decimal) {**

**int binary[32];**

**int index = 0;**

**if (decimal == 0) {**

**printf("Binary: 0\n");**

**return;**

**}**

**while (decimal > 0) {**

**binary[index] = decimal % 2;**

**decimal /= 2;**

**index++;**

**}**

**printf("Binary: ");**

**for (int i = index - 1; i >= 0; i--) {**

**printf("%d", binary[i]);**

**}**

**printf("\n");**

**}**

**int main() {**

**int choice, number;**

**printf("Choose the operation:\n");**

**printf("1. Convert Binary to Decimal\n");**

**printf("2. Convert Decimal to Binary\n");**

**printf("Enter your choice (1 or 2): ");**

**scanf("%d", &choice);**

**if (choice == 1)**

**{**

**printf("Enter a binary number: ");**

**scanf("%d", &number);**

**printf("Decimal: %d\n", binary\_to\_decimal(number));**

**}**

**else if (choice == 2)**

**{**

**printf("Enter a decimal number: ");**

**scanf("%d", &number);**

**decimal\_to\_binary(number);**

**}**

**else**

**{**

**printf("Invalid choice! Please enter 1 or 2.\n");**

**}**

**return 0;**

**}**

**Output**

**/tmp/zQ3HzYRTpf.o**

**Choose the operation:**

**1. Convert Binary to Decimal**

**2. Convert Decimal to Binary**

**Enter your choice (1 or 2): 2**

**Enter a decimal number: 13**

**Binary: 1101**

**=== Code Execution Successful ===**

**/tmp/2AXr4fEOW1.o**

**Choose the operation:**

**1. Convert Binary to Decimal**

**2. Convert Decimal to Binary**

**Enter your choice (1 or 2): 1**

**Enter a binary number: 1101**

**Decimal: 13**

**=== Code Execution Successful ===**

**QUESTION-8)**

**Input**

**#include <stdio.h>**

**int main() {**

**int i, j, n = 5;**

**for (i = 1; i <= n; i++)**

**{**

**for (j = 1; j <= i; j++)**

**{**

**if ((i + j) % 2 == 0) {**

**printf("1");**

**} else {**

**printf("0");**

**}**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/zWKJdFXv87.o**

**1**

**01**

**101**

**0101**

**10101**

**=== Code Execution Successful ===**

**QUESTION-9)**

**INPUT**

**#include <stdio.h>**

**int main() {**

**int i, j, n = 5;**

**for (i = 1; i <= n; i++)**

**{**

**for (j = 1; j <= i; j++)**

**{**

**if (j % 2 == 0)**

**{**

**printf("1");**

**} else {**

**printf("0");**

**}**

**}**

**for (j = 1; j <= 2 \* (n - i) - 1; j++)**

**{**

**printf(" ");**

**}**

**for (j = 1; j <= i; j++)**

**{**

**if (j % 2 == 0) {**

**printf("1");**

**} else {**

**printf("0");**

**}**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/PG26xoa9u4.o**

**0 0**

**01 01**

**010 010**

**0101 0101**

**0101001010**

**=== Code Execution Successful ===**

**QUESTION-10)**

**INPUT**

**#include <stdio.h>**

**long long factorial(int n) {**

**long long fact = 1;**

**for (int i = 1; i <= n; i++) {**

**fact \*= i;**

**}**

**return fact;**

**}**

**long long binomialCoefficient(int n, int r) {**

**return factorial(n) / (factorial(r) \* factorial(n - r));**

**}**

**int main() {**

**int n;**

**printf("Enter the number of rows for Pascal's Triangle: ");**

**scanf("%d", &n);**

**for (int i = 0; i < n; i++)**

**{**

**for (int space = 0; space < n - i - 1; space++)**

**{**

**printf(" ");**

**}**

**for (int j = 0; j <= i; j++)**

**{**

**printf("%lld ", binomialCoefficient(i, j));**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/Xh9gLldYwy.o**

**Enter the number of rows for Pascal's Triangle: 5**

**1**

**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

**=== Code Execution Successful ===**

**ASSIGNMENT-2**

**QUESTION-1:**

**INPUT**

**//Write a program to increase every student mark by 5 and then print the updated array.**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int marks[10],i;**

**for(i=0;i<10;i++)**

**{**

**printf("\n Enter the marks for student %d :",i);**

**scanf("%d",&marks[i]);**

**marks[i]=marks[i]+5;**

**printf(" MARKS :%d",marks[i]);**

**}**

**return 0;**

**}**

**OUTPUT**

**Enter the marks for student 0 :2**

**MARKS :7**

**Enter the marks for student 1 :8**

**MARKS :13**

**Enter the marks for student 2 :9**

**MARKS :14**

**Enter the marks for student 3 :6**

**MARKS :11**

**Enter the marks for student 4 :8**

**MARKS :13**

**Enter the marks for student 5 :5**

**MARKS :10**

**Enter the marks for student 6 :7**

**MARKS :12**

**Enter the marks for student 7 :4**

**MARKS :9**

**Enter the marks for student 8 :3**

**MARKS :8**

**Enter the marks for student 9 :7**

**MARKS :12**

**=== Code Execution Successful ===**

**QUESTION-2:**

**INPUT**

**//Write a program to print the grade of students as per their marks.(>=75--A grade, 74 to 60--B grade, 59 to 40--C grade below 40--D grade).**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int marks[10],i;**

**for(i=0;i<10;i++)**

**{**

**printf("\n Enter the marks for student %d :",i);**

**scanf("%d",&marks[i]);**

**int total=100;**

**if(marks[i]>=75)**

**{**

**printf("Grade A");**

**}**

**else if(marks[i]>=60)**

**{**

**printf("Grade B");**

**}**

**else if(marks[i]>=40)**

**{**

**printf("Grade C");**

**}**

**else**

**{**

**printf("Grade D");**

**}**

**}**

**}**

**return 0;**

**}**

**OUTPUT**

**Enter the marks for student 0 :56**

**Grade C**

**Enter the marks for student 1 :78**

**Grade A**

**Enter the marks for student 2 :45**

**Grade C**

**Enter the marks for student 3 :87**

**Grade A**

**Enter the marks for student 4 :99**

**Grade A**

**Enter the marks for student 5 :45**

**Grade C**

**Enter the marks for student 6 :99**

**Grade A**

**Enter the marks for student 7 :99**

**Grade A**

**Enter the marks for student 8 :60**

**Grade B**

**Enter the marks for student 9 :34**

**Grade D**

**=== Code Execution Successful ===**

**QUESTION-3:**

**INPUT**

**//Write a program to find who scored first "99" in an array marks.**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int i;**

**int flag=0;**

**int marks[10]={44,88,90,76,99,75,56,67,99,70};**

**for(i=0;i<10;i++)**

**{**

**if(marks[i]==99)**

**{flag=1;**

**printf("\nindex=%d \nmarks[i]=%d",i,marks[i]);**

**break;}**

**}**

**if(flag==0)**

**printf("NOBODY GOT 99");**

**return 0;**

**}**

**OUTPUT**

**index=4**

**marks[i]=99**

**=== Code Execution Successful ===**

**QUESTION-4:**

**INPUT**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int i;**

**int marks[10]={44,88,90,76,99,99,56,67,99,70};**

**for(i=0;i<10;i++)**

**{**

**if(marks[i]==99)**

**{**

**printf("\nindex=%d \nmarks[i]=%d",i,marks[i]);**

**}**

**}**

**return 0;**

**}**

**OUTPUT**

**index=4**

**marks[i]=99**

**index=5**

**marks[i]=99**

**index=8**

**marks[i]=99**

**=== Code Execution Successful ===**

**QUESTION-5:**

**INPUT**

**//Write a program to find the sum of all scores in Marks array.**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int max=5,sum=0;**

**int marks[max],i;**

**for(i=0;i<5;i++)**

**{**

**printf("\n Enter the marks for student %d :",i);**

**scanf("%d",&marks[i]);**

**}**

**for(i=0;i<5;i++)**

**{**

**sum=sum+marks[i];**

**}**

**printf(" SUM :%d",sum);**

**return 0;**

**}**

**OUTPUT**

**Enter the marks for student 0 :76**

**Enter the marks for student 1 :58**

**Enter the marks for student 2 :97**

**Enter the marks for student 3 :43**

**Enter the marks for student 4 :87**

**SUM :361**

**=== Code Execution Successful ===**

**QUESTION-6:**

**INPUT**

**//Write a program to find average score of the Marks array.**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int max=5,sum=0,avg=0;**

**int marks[max],i;**

**for(i=0;i<5;i++)**

**{**

**printf("\n Enter the marks for student %d :",i);**

**scanf("%d",&marks[i]);**

**}**

**for(i=0;i<5;i++)**

**{**

**sum=sum+marks[i];**

**avg=sum/5;**

**}**

**printf(" AVERAGE :%d",avg);**

**return 0;**

**}**

**OUTPUT**

**Enter the marks for student 0 :54**

**Enter the marks for student 1 :35**

**Enter the marks for student 2 :78**

**Enter the marks for student 3 :96**

**Enter the marks for student 4 :56**

**AVERAGE :63**

**=== Code Execution Successful ===**

**QUESTION-7:**

**INPUT**

**#include <stdio.h>**

**int main() {**

**int n;**

**printf("Enter the number of scores: ");**

**scanf("%d", &n);**

**int scores[n];**

**printf("Enter the scores:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &scores[i]);**

**}**

**printf("Results:\n");**

**for (int i = 0; i < n; i++)**

**{**

**if (scores[i] % 2 == 0)**

**{**

**printf("Score %d is Even\n", scores[i]);**

**} else {**

**printf("Score %d is Odd\n", scores[i]);**

**}**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/Z6W50nlVj5.o**

**Enter the number of scores: 3**

**Enter the scores:**

**8**

**9**

**7**

**Results:**

**Score 8 is Even**

**Score 9 is Odd**

**Score 7 is Odd**

**=== Code Execution Successful ===**

**QUESTION-8:**

**INPUT**

**#include <stdio.h>**

**int main() {**

**int n;**

**printf("Enter the number of students: ");**

**scanf("%d", &n);**

**int marks[n];**

**printf("Enter the marks of %d students:\n", n);**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &marks[i]);**

**}**

**int max = marks[0];**

**int min = marks[0];**

**for (int i = 1; i < n; i++)**

**{**

**if (marks[i] > max)**

**{**

**max = marks[i];**

**}**

**if (marks[i] < min)**

**{**

**min = marks[i];**

**}**

**}**

**printf("Maximum score: %d\n", max);**

**printf("Minimum score: %d\n", min);**

**return 0;**

**}**

**OUTPUT**

**/tmp/USGI7knChz.o**

**Enter the number of students: 6**

**Enter the marks of 6 students:**

**87**

**56**

**89**

**49**

**90**

**26**

**Maximum score: 90**

**Minimum score: 26**

**=== Code Execution Successful ===**

**QUESTION-9:**

**INPUT**

**#include <stdio.h>**

**int findPeak(int arr[], int n) {**

**if (n == 1 || arr[0] >= arr[1])**

**{**

**return 0;**

**}**

**if (arr[n-1] >= arr[n-2])**

**{**

**return n-1;**

**}**

**for (int i = 1; i < n-1; i++)**

**{**

**if (arr[i] >= arr[i-1] && arr[i] >= arr[i+1])**

**{**

**return i;**

**}**

**}**

**return -1;**

**}**

**int main() {**

**int n;**

**printf("Enter the number of elements: ");**

**scanf("%d", &n);**

**if (n <= 0)**

**{**

**printf("Array size must be greater than 0.\n");**

**return -1;**

**}**

**int arr[n];**

**printf("Enter the elements of the array:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**int peakIndex = findPeak(arr, n);**

**if (peakIndex != -1)**

**{**

**printf("Peak element is %d at index %d\n", arr[peakIndex], peakIndex);**

**} else {**

**printf("No peak element found.\n");**

**}**

**return 0;**

**}**

**Output**

**/tmp/B1Zq5PtJAn.o**

**Enter the number of elements: 5**

**Enter the elements of the array:**

**1**

**2**

**10**

**3**

**0**

**Peak element is 10 at index 2**

**=== Code Execution Successful ===**

**QUESTION-10:**

**INPUT**

**#include <stdio.h>**

**#include <stdbool.h>**

**bool isPrime(int num) {**

**if (num <= 1) return false;**

**for (int i = 2; i \* i <= num; i++)**

**{**

**if (num % i == 0) return false;**

**}**

**return true;**

**}**

**int main() {**

**int n;**

**printf("Enter the number of elements: ");**

**scanf("%d", &n);**

**if (n <= 0)**

**{**

**printf("Array size must be greater than 0.\n");**

**return -1;**

**}**

**int arr[n];**

**printf("Enter the elements of the array:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**int primeCount = 0;**

**for (int i = 0; i < n; i++)**

**{**

**if (isPrime(arr[i]))**

**{**

**primeCount++;**

**}**

**}**

**printf("The number of prime numbers in the array is: %d\n", primeCount);**

**return 0;**

**}**

**OUTPUT**

**/tmp/82nkv1IJD5.o**

**Enter the number of elements: 6**

**Enter the elements of the array:**

**2**

**6**

**8**

**7**

**11**

**13**

**The number of prime numbers in the array is: 4**

**=== Code Execution Successful ===**

**QUESTION-11:**

**INPUT**

**#include <stdio.h>**

**void printArray(int arr[], int n) {**

**printf("Array: ");**

**for (int i = 0; i < n; i++)**

**{**

**printf("%d ", arr[i]);**

**}**

**printf("\n");**

**}**

**void insertFront(int arr[], int \*n, int element) {**

**for (int i = \*n; i > 0; i--)**

**{**

**arr[i] = arr[i - 1];**

**}**

**arr[0] = element;**

**(\*n)++;**

**}**

**void insertAtPosition(int arr[], int \*n, int element, int pos) {**

**if (pos > \*n || pos < 0)**

**{**

**printf("Invalid position!\n");**

**return;**

**}**

**for (int i = \*n; i > pos; i--)**

**{**

**arr[i] = arr[i - 1];**

**}**

**arr[pos] = element;**

**(\*n)++;**

**}**

**void insertEnd(int arr[], int \*n, int element) {**

**arr[\*n] = element;**

**(\*n)++;**

**}**

**int main() {**

**int n, element, pos;**

**printf("Enter the number of elements in the array: ");**

**scanf("%d", &n);**

**int arr[n + 1];**

**printf("Enter the elements of the array:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**printf("Before Insert:\n");**

**printArray(arr, n);**

**printf("\nEnter the element to insert at the front: ");**

**scanf("%d", &element);**

**insertFront(arr, &n, element);**

**printf("After inserting at the front:\n");**

**printArray(arr, n);**

**printf("\nEnter the element to insert: ");**

**scanf("%d", &element);**

**printf("Enter the position to insert (0 to %d): ", n);**

**scanf("%d", &pos);**

**insertAtPosition(arr, &n, element, pos);**

**printf("After inserting at position %d:\n", pos);**

**printArray(arr, n);**

**printf("\nEnter the element to insert at the end: ");**

**scanf("%d", &element);**

**insertEnd(arr, &n, element);**

**printf("After inserting at the end:\n");**

**printArray(arr, n);**

**return 0;**

**}**

**OUTPUT**

**/tmp/cImMuFeJR1.o**

**Enter the number of elements in the array: 4**

**Enter the elements of the array:**

**1**

**2**

**3**

**4**

**Before Insert:**

**Array: 1 2 3 4**

**Enter the element to insert at the front: 1**

**After inserting at the front:**

**Array: 1 1 2 3 4**

**Enter the element to insert: 3**

**Enter the position to insert (0 to 5): 3**

**After inserting at position 3:**

**Array: 1 1 2 3 3 4**

**Enter the element to insert at the end: 5**

**After inserting at the end:**

**Array: 1 1 2 3 3 4 5**

**=== Code Execution Successful ===**

**QUESTION-12:**

**INPUT**

**#include <stdio.h>**

**void printArray(int arr[], int n) {**

**printf("Array: ");**

**for (int i = 0; i < n; i++)**

**{**

**printf("%d ", arr[i]);**

**}**

**printf("\n");**

**}**

**void deleteFront(int arr[], int \*n) {**

**for (int i = 0; i < \*n - 1; i++)**

**{**

**arr[i] = arr[i + 1];**

**}**

**(\*n)--;**

**}**

**void deleteAtPosition(int arr[], int \*n, int pos) {**

**if (pos < 0 || pos >= \*n)**

**{**

**printf("Invalid position!\n");**

**return;**

**}**

**for (int i = pos; i < \*n - 1; i++)**

**{**

**arr[i] = arr[i + 1];**

**}**

**(\*n)--;**

**}**

**void deleteEnd(int arr[], int \*n) {**

**(\*n)--;**

**}**

**int main() {**

**int n, pos;**

**printf("Enter the number of elements in the array: ");**

**scanf("%d", &n);**

**if (n <= 0)**

**{**

**printf("Array size must be greater than 0.\n");**

**return -1;**

**}**

**int arr[n];**

**printf("Enter the elements of the array:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**printf("Before Delete:\n");**

**printArray(arr, n);**

**deleteFront(arr, &n);**

**printf("After deleting the front element:\n");**

**printArray(arr, n);**

**printf("\nEnter the position to delete (0 to %d): ", n - 1);**

**scanf("%d", &pos);**

**deleteAtPosition(arr, &n, pos);**

**printf("After deleting the element at position %d:\n", pos);**

**printArray(arr, n);**

**deleteEnd(arr, &n);**

**printf("After deleting the last element:\n");**

**printArray(arr, n);**

**return 0;**

**}**

**OUTPUT**

**/tmp/8yRCRTFGWy.o**

**Enter the number of elements in the array: 5**

**Enter the elements of the array:**

**1**

**2**

**3**

**4**

**5**

**Before Delete:**

**Array: 1 2 3 4 5**

**After deleting the front element:**

**Array: 2 3 4 5**

**Enter the position to delete (0 to 3): 3**

**After deleting the element at position 3:**

**Array: 2 3 4**

**After deleting the last element:**

**Array: 2 3**

**=== Code Execution Successful ===**

**QUESTION-13:**

**INPUT**

**#include <stdio.h>**

**void rotateArray(int arr[], int n) {**

**int last = arr[n - 1];**

**for (int i = n - 1; i > 0; i--)**

**{**

**arr[i] = arr[i - 1];**

**}**

**arr[0] = last;**

**}**

**void printArray(int arr[], int n) {**

**for (int i = 0; i < n; i++)**

**{**

**printf("%d ", arr[i]);**

**}**

**printf("\n");**

**}**

**int main() {**

**int n;**

**printf("Enter the number of elements in the array: ");**

**scanf("%d", &n);**

**if (n <= 0)**

**{**

**printf("Array size must be greater than 0.\n");**

**return -1;**

**}**

**int arr[n];**

**printf("Enter the elements of the array:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**printf("Original Array:\n");**

**printArray(arr, n);**

**rotateArray(arr, n);**

**printf("Array after rotating clockwise by one position:\n");**

**printArray(arr, n);**

**return 0;**

**}**

**OUTPUT**

**/tmp/n4d2LK92Bb.o**

**Enter the number of elements in the array: 7**

**Enter the elements of the array:**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**Original Array:**

**1 2 3 4 5 6 7**

**Array after rotating clockwise by one position:**

**7 1 2 3 4 5 6**

**=== Code Execution Successful ===**

**QUESTION-14:**

**INPUT**

**#include <stdio.h>**

**void printDuplicates(int arr[], int n) {**

**int foundDuplicate = 0;**

**for (int i = 0; i < n; i++)**

**{**

**if (arr[i] < 0)**

**{**

**continue;**

**}**

**for (int j = i + 1; j < n; j++)**

**{**

**if (arr[i] == arr[j])**

**{**

**printf("%d ", arr[i]);**

**foundDuplicate = 1;**

**break;**

**}**

**}**

**}**

**if (!foundDuplicate)**

**{**

**printf("-1\n");**

**}**

**}**

**int main() {**

**int n;**

**printf("Enter the number of elements in the array: ");**

**scanf("%d", &n);**

**if (n <= 0)**

**{**

**printf("Array size must be greater than 0.\n");**

**return -1;**

**}**

**int arr[n];**

**printf("Enter the elements of the array:\n");**

**for (int i = 0; i < n; i++)**

**{**

**scanf("%d", &arr[i]);**

**}**

**printf("Duplicates in the array are: ");**

**printDuplicates(arr, n);**

**return 0;**

**}**

**OUTPUT**

**/tmp/wCphiA6KiO.o**

**Enter the number of elements in the array: 10**

**Enter the elements of the array:**

**1**

**2**

**2**

**3**

**4**

**5**

**5**

**6**

**7**

**7**

**Duplicates in the array are: 2 5 7**

**=== Code Execution Successful ===**

**ASSIGNMENT-3**

**INPUT**

**#include <stdio.h>**

**#include <math.h>**

**void Addition(){**

**int a,b;**

**printf("Enter first number: ");**

**scanf("%d",&a);**

**printf("Enter second number: ");**

**scanf("%d",&b);**

**printf("SUM %d",a+b);**

**}**

**void Subtraction(){**

**int a,b;**

**printf("Enter the first number: ");**

**scanf("%d",&a);**

**printf("Enter the second number: ");**

**scanf("%d",&b);**

**if(a>b)**

**printf("Subtraction %d",a-b);**

**else**

**printf("Subtraction %d",b-a);**

**}**

**void Multiplication(){**

**int a,b;**

**printf("Enter first number: ");**

**scanf("%d",&a);**

**printf("Enter second number: ");**

**scanf("%d",&b);**

**printf("Multiplication %d",a\*b);**

**}**

**void Division(){**

**int a,b;**

**printf("Enter first number: ");**

**scanf("%d",&a);**

**printf("Enter second number: ");**

**scanf("%d",&b);**

**if(a>b)**

**printf("Division %d",a/b);**

**else**

**printf("Division %d",b/a);**

**}**

**void Log(){**

**double a;**

**printf("Enter the value of a number to find the logarithm(base e): ");**

**scanf("%lf",&a);**

**if(a>0)**

**printf("%lf",log(a));**

**else**

**printf("ERROR! Logarithm of non-positive numbers are not defined.\n");**

**}**

**void Squareroot(){**

**double a;**

**printf("Enter a number to find the square root: ");**

**scanf("%lf",&a);**

**if(a>=0)**

**printf("%lf",sqrt(a));**

**else**

**printf("ERROR! Square root of a negative number is not defined.\n");**

**}**

**int main(){**

**printf("WELCOME TO SIMPLE CALCULATOR \n It can perform\n1) Addition\n2)Subtraction\n3)Multiplication\n4)Division\n5)Logarithm\n6)Square Root\n");**

**printf("Enter your Choice: ");**

**int choice;**

**scanf("%d",&choice);**

**switch(choice)**

**{**

**case 1:**

**Addition();**

**break;**

**case 2:**

**Subtraction();**

**break;**

**case 3:**

**Multiplication();**

**break;**

**case 4:**

**Division();**

**break;**

**case 5:**

**Log();**

**break;**

**case 6:**

**Squareroot();**

**break;**

**default:**

**printf("Entered the wrong choice");**

**break;**

**}**

**return 0;**

**}**

**Output**

**/tmp/HQACv36C4F.o**

**WELCOME TO SIMPLE CALCULATOR**

**It can perform**

**1) Addition**

**2)Subtraction**

**3)Multiplication**

**4)Division**

**5)Logarithm**

**6)Square Root**

**Enter your Choice: 3**

**Enter first number: 8**

**Enter second number: 9**

**Multiplication 72**

**=== Code Execution Successful ===**

**ASSIGNMENT-4**

**INPUT**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <time.h>**

**#include <string.h>**

**#include <ctype.h>**

**const char\* get\_computer\_choice() {**

**int random\_number = rand() % 101;**

**if (random\_number >= 0 && random\_number <= 33)**

**{**

**return "Rock";**

**} else if (random\_number >= 34 && random\_number <= 66)**

**{**

**return "Paper";**

**} else {**

**return "Scissors";**

**}**

**}**

**const char\* determine\_winner(const char\* user\_choice, const char\* computer\_choice){**

**if (strcmp(user\_choice, computer\_choice) == 0)**

**{**

**return "It's a tie.";**

**} else if ((strcmp(user\_choice, "Rock") == 0 && strcmp(computer\_choice, "Scissors") == 0) ||**

**(strcmp(user\_choice, "Paper") == 0 && strcmp(computer\_choice, "Rock") == 0) ||**

**(strcmp(user\_choice, "Scissors") == 0 && strcmp(computer\_choice, "Paper") == 0))**

**{**

**return "You win.";**

**} else {**

**return "Computer wins.";**

**}**

**}**

**int main(){**

**srand(time(0));**

**char user\_choice[10];**

**printf("Enter your choice (Rock, Paper, Scissors): ");**

**scanf("%s", user\_choice);**

**for (int i = 0; user\_choice[i]; i++)**

**{**

**user\_choice[i] = tolower(user\_choice[i]);**

**}**

**user\_choice[0] = toupper(user\_choice[0]);**

**if (strcmp(user\_choice, "Rock") != 0 && strcmp(user\_choice, "Paper") != 0 && strcmp(user\_choice, "Scissors") != 0)**

**{**

**printf("Invalid choice! Please choose Rock, Paper, or Scissors.\n");**

**return 1;**

**}**

**const char\* computer\_choice = get\_computer\_choice();**

**printf("\nYour choice: %s\n", user\_choice);**

**printf("Computer's choice: %s\n", computer\_choice);**

**printf("%s\n", determine\_winner(user\_choice, computer\_choice));**

**return 0;**

**}**

**OUTPUT**

**/tmp/upqxBTeB0Q.o**

**Enter your choice (Rock, Paper, Scissors): SCISSORS**

**Your choice: Scissors**

**Computer's choice: Paper**

**You win.**

**=== Code Execution Successful ===**

**ASSIGNMENT-5**

**INPUT**

**#include <stdio.h>**

**#include <string.h>**

**#include <ctype.h>**

**int main() {**

**char word[] = "coding";**

**char guessedWord[20];**

**int wordLength = strlen(word);**

**int attempts = 3;**

**int correctGuesses = 0;**

**char guess;**

**for (int i = 0; i < wordLength; i++)**

**{**

**guessedWord[i] = '\_';**

**}**

**guessedWord[wordLength] = '\0';**

**printf("Welcome to Hangman!\n");**

**printf("You have %d attempts to guess the word.\n", attempts);**

**while (attempts > 0 && correctGuesses < wordLength)**

**{**

**printf("Current word: %s\n", guessedWord);**

**printf("Enter your guess: ");**

**scanf(" %c", &guess);**

**guess = tolower(guess);**

**int found = 0;**

**for (int i = 0; i < wordLength; i++)**

**{**

**if (word[i] == guess && guessedWord[i] == '\_')**

**{**

**guessedWord[i] = guess;**

**correctGuesses++;**

**found = 1;**

**}**

**}**

**if (!found)**

**{**

**attempts--;**

**printf("Wrong guess! Attempts remaining: %d\n",**

**attempts);**

**} else {**

**printf("Good guess!\n");**

**}**

**}**

**if (correctGuesses == wordLength)**

**{**

**printf("Congratulations! You guessed the word: %s\n", word);**

**printf("The Man survives!\n");**

**} else {**

**printf("You ran out of attempts! The word was: %s\n", word);**

**printf("The Man is hanged!\n");**

**}**

**return 0;**

**}**

**OUTPUT**

**/tmp/KDmuZipTi7.o**

**Welcome to Hangman!**

**You have 3 attempts to guess the word.**

**Current word: \_\_\_\_\_\_**

**Enter your guess: C**

**Good guess!**

**Current word: c\_\_\_\_\_**

**Enter your guess: R**

**Wrong guess! Attempts remaining: 2**

**Current word: c\_\_\_\_\_**

**Enter your guess: O**

**Good guess!**

**Current word: co\_\_\_\_**

**Enter your guess: I**

**Good guess!**

**Current word: co\_i\_\_**

**Enter your guess: G**

**Good guess!**

**Current word: co\_i\_g**

**Enter your guess: F**

**Wrong guess! Attempts remaining: 1**

**Current word: co\_i\_g**

**Enter your guess: U**

**Wrong guess! Attempts remaining: 0**

**You ran out of attempts! The word was: coding**

**The Man is hanged!**

**=== Code Execution Successful ===**

**ASSIGNMENT-6**

**INPUT**

**#include <stdio.h>**

**#define SIZE 3**

**void printBoard(char board[SIZE][SIZE]);**

**int checkWin(char board[SIZE][SIZE]);**

**int isDraw(char board[SIZE][SIZE]);**

**int main() {**

**char board[SIZE][SIZE] = {{' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', ' '}};**

**int row, col, turn = 0;**

**char player = 'X';**

**while (1)**

**{**

**printBoard(board);**

**printf("Player %c, enter row (1-3) and column (1-3): ", player);**

**scanf("%d %d", &row, &col);**

**row--; col--;**

**if (row >= 0 && row < SIZE && col >= 0 && col < SIZE && board[row][col] == ' ')**

**{**

**board[row][col] = player;**

**turn++;**

**if (checkWin(board))**

**{**

**printBoard(board);**

**printf("Player %c wins!\n", player);**

**break;**

**}**

**if (isDraw(board))**

**{**

**printBoard(board);**

**printf("It's a draw!\n");**

**break;**

**}**

**player = (player == 'X') ? 'O' : 'X';**

**} else {**

**printf("Invalid move. Try again.\n");**

**}**

**}**

**return 0;**

**}**

**void printBoard(char board[SIZE][SIZE]) {**

**printf("\n");**

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

**{**

**for (int j = 0; j < SIZE; j++)**

**{**

**printf(" %c ", board[i][j]);**

**if (j < SIZE - 1) printf("|");**

**}**

**printf("\n");**

**if (i < SIZE - 1) printf("---|---|---\n");**

**}**

**printf("\n");**

**}**

**int checkWin(char board[SIZE][SIZE]) {**

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

**{**

**if ((board[i][0] == board[i][1] && board[i][1] == board[i][2] && board[i][0] != ' ') ||**

**(board[0][i] == board[1][i] && board[1][i] == board[2][i] && board[0][i] != ' '))**

**{**

**return 1;**

**}**

**}**

**if ((board[0][0] == board[1][1] && board[1][1] == board[2][2] && board[0][0] != ' ') ||**

**(board[0][2] == board[1][1] && board[1][1] == board[2][0] && board[0][2] != ' '))**

**{**

**return 1;**

**}**

**return 0;**

**}**

**int isDraw(char board[SIZE][SIZE]) {**

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

**{**

**for (int j = 0; j < SIZE; j++)**

**{**

**if (board[i][j] == ' ') return 0;**

**}**

**}**

**return 1;**

**}**

**Output**

**/tmp/GzYCUTb4gx.o**

**| |**

**---|---|---**

**| |**

**---|---|---**

**| |**

**Player X, enter row (1-3) and column (1-3): 2 2**

**| |**

**---|---|---**

**| X |**

**---|---|---**

**| |**

**Player O, enter row (1-3) and column (1-3): 3 3**

**| |**

**---|---|---**

**| X |**

**---|---|---**

**| | O**

**Player X, enter row (1-3) and column (1-3): 1 1**

**X | |**

**---|---|---**

**| X |**

**---|---|---**

**| | O**

**Player O, enter row (1-3) and column (1-3): 1 3**

**X | | O**

**---|---|---**

**| X |**

**---|---|---**

**| | O**

**Player X, enter row (1-3) and column (1-3): 2 3**

**X | | O**

**---|---|---**

**| X | X**

**---|---|---**

**| | O**

**Player O, enter row (1-3) and column (1-3): 3 1**

**X | | O**

**---|---|---**

**| X | X**

**---|---|---**

**O | | O**

**Player X, enter row (1-3) and column (1-3): 2 1**

**X | | O**

**---|---|---**

**X | X | X**

**---|---|---**

**O | | O**

**Player X wins!**

**=== Code Execution Successful ===**