**VISHAW BANSAL**

**1024030356**

**ASSIGNMENT 2**

**Answer 1**

**#include <iostream>**

**using namespace std;**

**int linearSearch(int arr[], int n, int key) {**

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

**if (arr[i] == target) {**

**return i;**

**}**

**}**

**return -1;**

**}**

**int binarySearch(int arr[], int n, int key) {**

**int first = 0, high = n - 1;**

**while (first <= high) {**

**int mid = (first + high) / 2;**

**if (arr[mid] == key) {**

**return mid;**

**} else if (arr[mid] < key) {**

**first = mid + 1;**

**} else {**

**high = mid - 1;**

**}**

**}**

**return -1;**

**}**

**int main() {**

**int arr[] = {2, 5, 8, 12, 16, 23, 38, 45, 56, 72, 91};**

**int n = sizeof(arr) / sizeof(arr[0]);**

**int x = 23;**

**cout << "Array: ";**

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

**cout << arr[i] << " ";**

**}**

**cout << endl;**

**cout << "Element you wanted to search: " << x << endl;**

**int res1 = linearSearch(arr, n, x);**

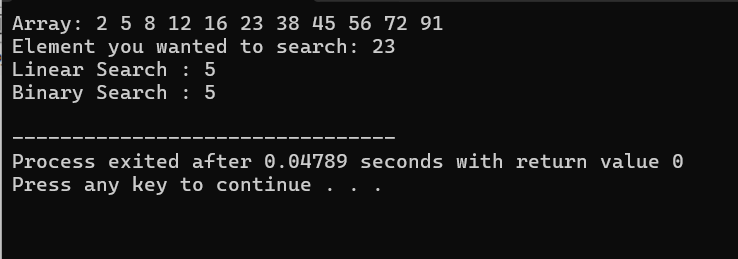
**cout << "Linear Search : " << res1 << endl;**

**int res2 = binarySearch(arr, n, x);**

**cout << "Binary Search : " << res2 << endl;**

**return 0;**

**}**

****

**Answer 2**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int arr[] = {5,15,20,10,4,8,11};**

**int n = 7;**

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

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

**if (arr[j] > arr[j+1]) {**

**int temp = arr[j];**

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

**arr[j+1] = temp;**

**}**

**}**

**}**

**cout << "bubble sorted array: ";**

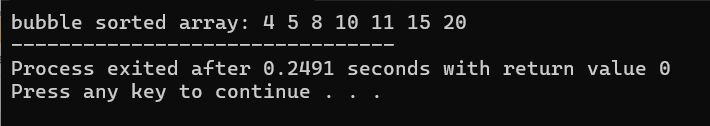
**for (int i = 0; i < n; i++) {**

**cout << arr[i] << " ";**

**}**

**return 0;**

**}**

****

**Answer 3**

**//Answer for 3(a)**

**#include <iostream>**

**using namespace std;**

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

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

**if (arr[i] != i+1) {**

**return i+1;**

**}**

**}**

**return n;**

**}**

**int main() {**

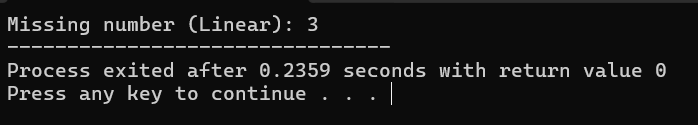
**int arr[] = {1, 2, 3, 5, 6};**

**int n = 6;**

**cout << "Missing number (Linear): " << findMissingLinear(arr, n);**

**return 0;**

**}**

****

**// answer for 3(b)**

**#include <iostream>**

**using namespace std;**

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

**int low = 0, high = n-2; // array size is n-1**

**while (low <= high) {**

**int mid = (low + high) / 2;**

**if (arr[mid] == mid + 1) {**

**low = mid + 1;**

**} else {**

**high = mid - 1;**

**}**

**}**

**return low + 1;**

**}**

**int main() {**

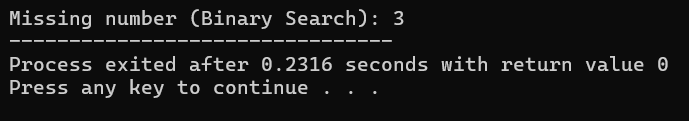
**int arr[] = {1, 2, 3, 5, 6};**

**int n = 6;**

**cout << "Missing number (Binary Search): " << findMissingBinary(arr, n);**

**return 0;**

**}**

****

**Answer 4**

**//Answer (a)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**char str1[100] = "Vishaw ";**

**char str2[50] = "Bansal";**

**int i = 0, j = 0;**

**while (str1[i] != '\0') {**

**i++;**

**}**

**while (str2[j] != '\0') {**

**str1[i] = str2[j];**

**i++;**

**j++;**

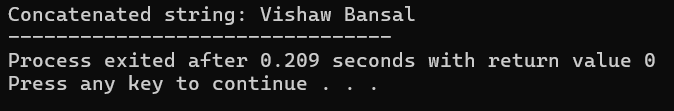
**}**

**str1[i] = '\0';**

**cout << "Concatenated string: " << str1;**

**return 0;**

**}**

****

**//Answer (b)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**char str[50] = "vishaw bansal";**

**int len = 0;**

**while (str[len] != '\0') {**

**len++;**

**}**

**cout << "Reversed string: ";**

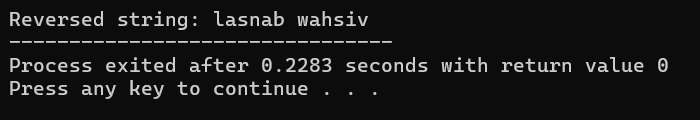
**for (int i = len-1; i >= 0; i--) {**

**cout << str[i];**

**}**

**return 0;**

**}**

****

**Answer (c)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**char str[50] = "vishaw bansal";**

**char result[50];**

**int i = 0, j = 0;**

**while (str[i] != '\0') {**

**char c = str[i];**

**if (!(c=='a'||c=='e'||c=='i'||c=='o'||c=='u'||**

**c=='A'||c=='E'||c=='I'||c=='O'||c=='U')) {**

**result[j++] = c;**

**}**

**i++;**

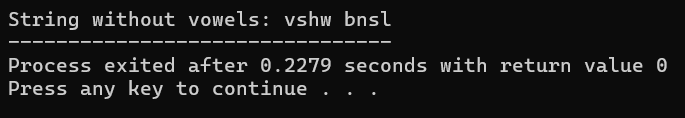
**}**

**result[j] = '\0';**

**cout << "String without vowels: " << result;**

**return 0;**

**}**

****

**Answer (d)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**char str[5][20] = {"banana", "apple", "mango", "cherry", "pear"};**

**char temp[20];**

**for (int i = 0; i < 4; i++) {**

**for (int j = 0; j < 4 - i; j++) {**

**int k = 0;**

**while (str[j][k] != '\0' && str[j+1][k] != '\0' && str[j][k] == str[j+1][k]) {**

**k++;**

**}**

**if (str[j][k] > str[j+1][k]) {**

**int p = 0;**

**while (true) {**

**temp[p] = str[j][p];**

**str[j][p] = str[j+1][p];**

**str[j+1][p] = temp[p];**

**if (str[j][p] == '\0' && str[j+1][p] == '\0')**

**break;**

**p++;**

**}**

**}**

**}**

**}**

**cout << "Strings in alphabetical order:\n";**

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

**cout << str[i] << endl;**

**}**

**return 0;**

**}**

**// Answer (e)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**char ch = 'A';**

**if (ch >= 'A' && ch <= 'Z') {**

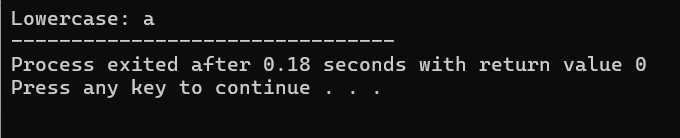
**ch = ch + 32;**

**}**

**cout << "Lowercase: " << ch;**

**return 0;**

**}**

****

**Answer 5**

**// Answer (a)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n = 4;**

**int dia[4];**

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

**cin >> dia[i];**

**}**

**// print full matrix**

**cout << "diagonal Matrix:\n";**

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

**for (int j = 0; j < n; j++) {**

**if (i == j) cout << dia[i] << " ";**

**else cout << 0 << " ";**

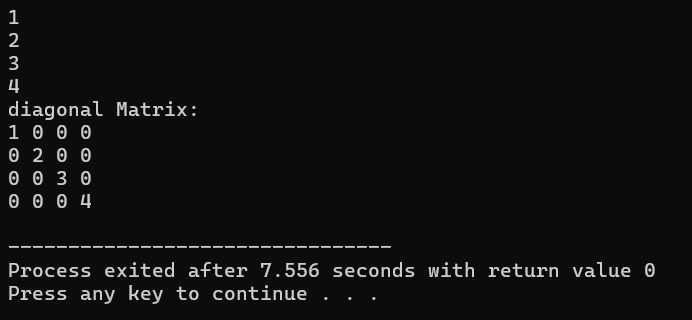
**}**

**cout << endl;**

**}**

**return 0;**

**}**

****

**//Answer (b)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n;**

**cout << "Enter size of matrix (n x n): ";**

**cin >> n;**

**int size = 3\*n - 2;**

**int tri[size];**

**cout << "Enter elements of tri-diagonal matrix :\n";**

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

**cin >> tri[i];**

**}**

**cout << "\nTri-diagonal Matrix:\n";**

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

**for (int j = 0; j < n; j++) {**

**if (i == j) {**

**cout << tri[i] << " ";**

**} else if (i == j+1) {**

**cout << tri[n + j] << " ";**

**} else if (i+1 == j) {**

**cout << tri[2\*n - 1 + i] << " ";**

**} else {**

**cout << 0 << " ";**

**}**

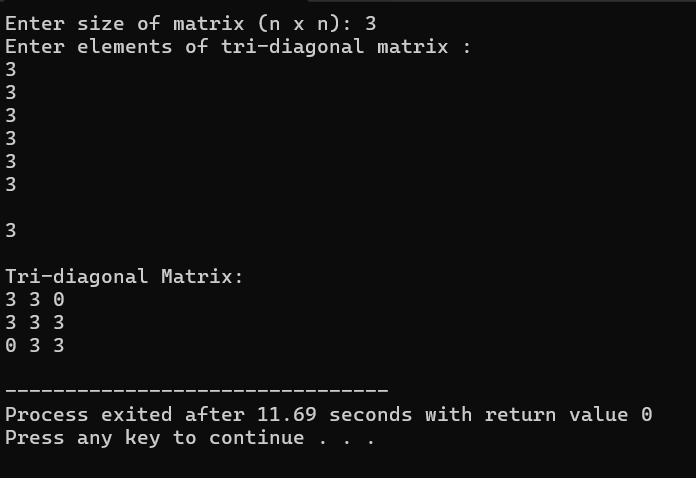
**}**

**cout << endl;**

**}**

**return 0;**

**}**

****

**//Answer(c)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n;**

**cout << "Enter size of matrix: ";**

**cin >> n;**

**int size = n\*(n+1)/2;**

**int lower[size];**

**cout << "Enter elements of lower triangular matrix:\n";**

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

**cin >> lower[i];**

**}**

**cout << "\nLower Triangular Matrix:\n";**

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

**for (int j = 0; j < n; j++) {**

**if (i >= j) {**

**int index = i\*(i+1)/2 + j;**

**cout << lower[index] << " ";**

**} else {**

**cout << 0 << " ";**

**}**

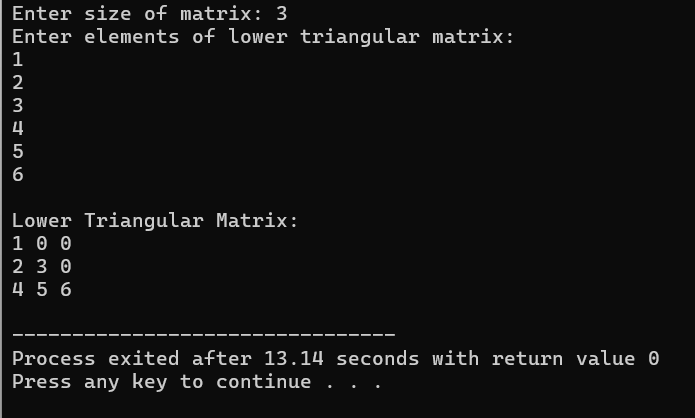
**}**

**cout << endl;**

**}**

**return 0;**

**}**

****

**//Answer(d)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n;**

**cout << "Enter size of matrix : ";**

**cin >> n;**

**int size = n\*(n+1)/2;**

**int upper[size];**

**cout << "Enter elements of upper triangular matrix:\n";**

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

**cin >> upper[i];**

**}**

**cout << "\nUpper Triangular Matrix:\n";**

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

**for (int j = 0; j < n; j++) {**

**if (i <= j) {**

**int index = (i\*n - (i\*(i-1))/2) + (j-i);**

**cout << upper[index] << " ";**

**} else {**

**cout << 0 << " ";**

**}**

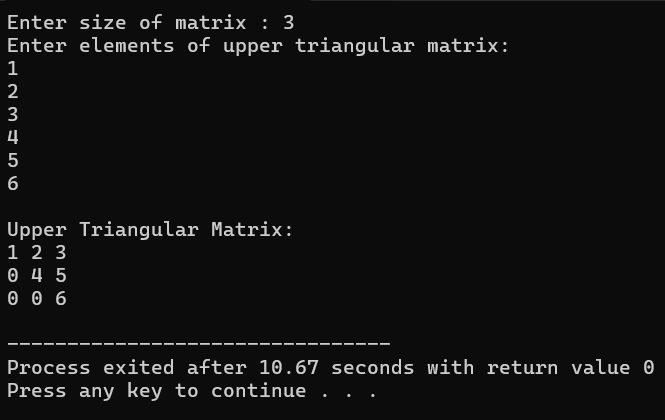
**}**

**cout << endl;**

**}**

**return 0;**

**}**

****

**//Answer(e)**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n;**

**cout << "Enter size of matrix : ";**

**cin >> n;**

**int size = n\*(n+1)/2;**

**int sym[size];**

**cout << "Enter elements :\n";**

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

**cin >> sym[i];**

**}**

**cout << "\nSymmetric Matrix:\n";**

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

**for (int j = 0; j < n; j++) {**

**if (i >= j) {**

**int index = i\*(i+1)/2 + j;**

**cout << sym[index] << " ";**

**} else {**

**int index = j\*(j+1)/2 + i;**

**cout << sym[index] << " ";**

**}**

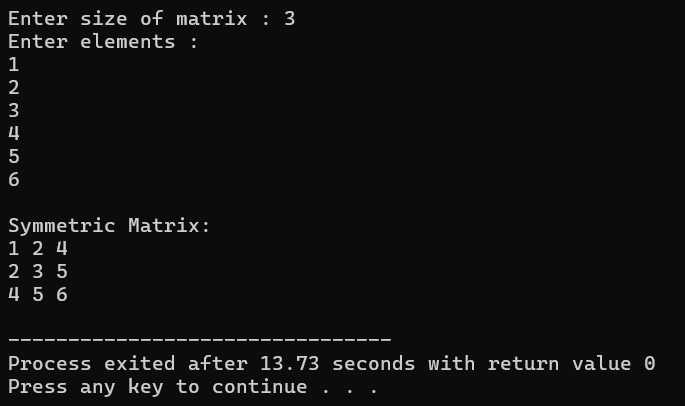
**}**

**cout << endl;**

**}**

**return 0;**

**}**

****

**Answer 6..**

**Answer 7…**

**Answer 8**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n;**

**cout << "Enter size of array: ";**

**cin >> n;**

**int arr[n];**

**cout << "Enter " << n << " elements:\n";**

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

**cin >> arr[i];**

**}**

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

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

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

**max = arr[i];**

**}**

**int hash[max + 1] = {0};**

**int distinct = 0;**

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

**if (hash[arr[i]] == 0) {**

**distinct++;**

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

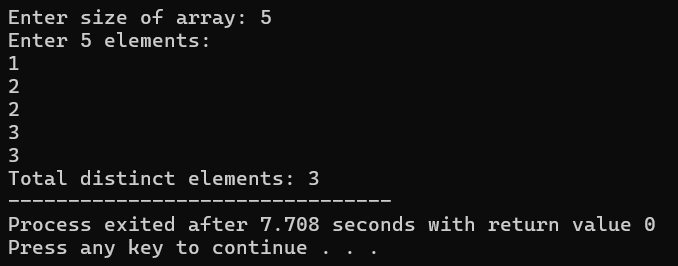
**}**

**}**

**cout << "Total distinct elements: " << distinct;**

**return 0;**

**}**

****