**Practical 8 – refer to Topics15 and 16**

**Part A(Understanding Concepts)**

1. Write statements for the following:
2. Declare an array that can store 10 integers.

Int arr[10];

1. Declare an array that can store 50 characters.

Char arr[50];

1. Declare an array that can store 15 values of type double.  
   double arr[15]
2. Given the following array declaration: int numbers[6];

Write statements for the following:

1. Assign the values 2, 50, and 17 to the first 3 elements.  
   Numbers [0] = 2;

Numbers [1] = 50;

Numbers [2] = 17;

1. Assign the values 8, 63, and 45 to the last 3 elements.

Numbers [3] = 8;

Numbers [4] = 63;

Numbers [5] = 45;

1. Decrease the value of the 5th element by 50.

Numbers [4] = numbers [4] – 50; OR numbers [4] -= 50;

1. Display the values of all the elements using a loop.

For (int i = 0; I < 6; I ++)

Cout << numbers [i] << endl;

1. Multiply the values of all the elements by 10 using a loop.

For (int I = 0; I < 6 ; I ++)

Numbers [i] \*= 10;

1. What is the output of the following program?

|  |
| --- |
| #include <iostream>  using namespace std;  int main(void )  {  int list1[5], list2[5];  for (int i = 0; i < 5; i++)  list1[i] = i;  for (int i = 0; i < 5; i++)  list2[i] = list1[4 – i];  for (int i = 0; i < 5; i++)  cout<< list2[i] << endl;  return 0;  } |

List1 = {0,1,2,3,4}

List2 = {4,3,2,1,0}

Display :

4

3

2

1

0

1. Write a function that has an array of characters and its size as parameters. The function initialises all the elements of the array to ‘\*’.

Void initilialize ( char letters [] , int size)

{  
for (int I = 0; I < size; i++)  
letters [i] = ‘\*’;  
}

**Part B (Programming Exercises)**

1. Write a C++ program with the functions described as follows:
2. A function that reads 10 integers entered by the user and stores them in an array.
3. A function that displays all the integers in the array.
4. A function that computes and returns the total of the integers in the array.
5. A main function that calls the functions that you write for part a, b and c.

Note that the functions in part a, b and c take two arguments: *an array of integers* and *an integer for the size of the array*.

#include <iostream>

#include <iomanip>

using namespace std;

void read(int numbers[],int size);

void display(int numbers[], int size);

int main(void)

{

int numbers[5];

cout << "Enter 5 numbers: ";

read(numbers, 5);

display(numbers, 5);

return 0;

}

void read(int numbers[], int size)

{

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

cin >> numbers[i];

}

void display(int numbers[], int size)

{

cout << "The numbers are: ";

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

cout << numbers[i] << " ";

cout << endl;

}

1. Modify the program in question 1 to add functions *find\_largest* and *find\_smallest* which find and return the largest and smallest numbers in the array respectively.

#include <iostream>

#define ARRAY\_SIZE 5

using namespace std;

int findTotal(int arr[], int size);

int findLargest(int arr[], int size);

int findSmallest(int arr[], int size);

int main()

{

int numbers[ARRAY\_SIZE];

int i, total, largest, smallest;

cout << "Enter " << ARRAY\_SIZE << " numbers: ";

for (i = 0; i < ARRAY\_SIZE; i++)

cin >> numbers[i];

total = findTotal(numbers, ARRAY\_SIZE);

cout << "Total is " << total << endl;

largest = findLargest(numbers, ARRAY\_SIZE);

smallest = findSmallest(numbers, ARRAY\_SIZE);

cout << "Largest is " << largest << endl;

cout << "Smallest is " << smallest << endl;

return 0;

}

int findTotal(int arr[], int size)

{

int i, total;

total = 0;

for (i = 0; i < size; i++)

total += arr[i];

return total;

}

int findLargest(int arr[], int size)

{

int i;

int largest = arr[0];

for (i = 1; i < size; i++)

if (largest < arr[i])

largest = arr[i];

return largest;

}

int findSmallest(int arr[], int size)

{

int i;

int smallest = arr[0];

for (i = 1; i < size; i++)

if (smallest > arr[i])

smallest = arr[i];

return smallest;

}

1. Write a function to test if every element of an array is equal to its corresponding element in another array. Let’s say we have two arrays, A and B. The function must check if A[0] is equal to B[0], A[1] is equal to B[1], and so forth. Your function is to return true if all elements are equal and return false if at least one element is not equal. The function header should look like this:

bool compare(int x[ ], int y[ ], int size)

where *size* is the size of the arrays. [*Note: Use a while loop instead of a for loop in your function. Also use a logical variable is\_equal.*]

Use the following program to test your function:

|  |
| --- |
| #include <iostream>  #define ARRAY\_SIZE 5  using namespace std;  bool compare(int x[], int y[], int size);  int main(void)  {  int a1[ARRAY\_SIZE] = {1, 2, 3, 4, 5};  int a2[ARRAY\_SIZE] = {1, 2, 3, 4, 5};  int a3[ARRAY\_SIZE] = {1, 1, 1, 1, 1};  if ( compare(a1, a2, ARRAY\_SIZE) )  cout<<"Arrays a1 and a2 are equal\n";  else  cout<<"Arrays a1 and a2 are not equal\n";  if ( compare(a1, a3, ARRAY\_SIZE) )  cout<<"Arrays a1 and a3 are equal\n";  else  cout<<"Arrays a1 and a3 are not equal\n";  return 0;  } |

#include <iostream>

#define ARRAY\_SIZE 5

using namespace std;

bool compare(int x[], int y[], int size);

int main(void)

{

int a1[ARRAY\_SIZE] = { 1, 2, 3, 4, 5 };

int a2[ARRAY\_SIZE] = { 1, 2, 3, 4, 5 };

int a3[ARRAY\_SIZE] = { 1, 1, 1, 1, 1 };

if (compare(a1, a2, ARRAY\_SIZE))

cout << "Arrays a1 and a2 are equal\n";

else

cout << "Arrays a1 and a2 are not equal\n";

if (compare(a1, a3, ARRAY\_SIZE))

cout << "Arrays a1 and a3 are equal\n";

else

cout << "Arrays a1 and a3 are not equal\n";

return 0;

}

bool compare(int x[], int y[], int size)

{

int i;

bool isEqual = true; //assume true

i = 0;

while (i < size && isEqual)

{

if (x[i] != y[i])

isEqual = false; //set to false

else

i++;

}

return isEqual;

}

1. Write a program that asks the user how many times he wants to roll a dice. Then use the function *rand* to generate the outcome of each roll.

Sample run:

How many times you want to roll a dice? 12

The outcomes of 12 rolls are: 3 6 4 1 6 4 2 1 1 5 2 5

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int main(void)

{

int count, rand\_no;

cout << "How many times you wanna roll the dice?!? ";

cin >> count;

srand(time(NULL));

cout << "The outcomes of the rolls are: ";

for (int i = 0; i < count; i++)

{

rand\_no = rand()%6+1 ;

cout << rand\_no << " ";

}

cout << endl;

return 0;

}

1. Modify the program in question 4 to display also the frequencies of the outcomes in a table as follows.

Outcome Frequency

======= ========

1 xx

2 xx

3 xx

4 xx

5 xx

6 xx

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int main(void)

{

int count, rand\_no, freq[6] = { 0 };

cout << "How many times you wanna roll the dice?!? ";

cin >> count;

srand(time(NULL));

cout << "The outcomes of the rolls are: ";

for (int i = 0; i < count; i++)

{

rand\_no = rand()%6+1 ;

cout << rand\_no << " ";

freq[rand\_no - 1]++;

}

cout << endl;

cout << "\n\nOutcome\tFrequency\n";

cout << "=======\t===========\n";

for (int i = 0; i < 6; i++)

cout << i + 1 << "\t" << freq[i] << endl;

return 0;

}

**Part C (Self-Review / Revision)**

1. What is the general format for an array declaration?
2. How do you access an element of an array?
3. Suppose function main has an array. If function main calls function f1 and passes the whole array, does function f1 access the same array or a copy of the array?

**Part D (Practice Exercises)**

1. Statisticians use many different algorithms to process numbers including the arithmetic average, geometric mean and harmonic mean.

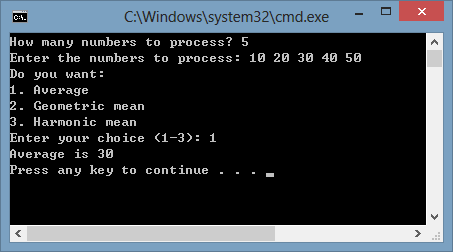
Given a set of *n* numbers *x*1, *x*2,*x*3, …,*x*n-1, *x*n, the arithmetic average is defined by the following formula:

The geometric mean is defined by the following formula:

The harmonic mean is defined by the following formula:

Write a program that reads a series of numbers and asks the user whether to calculate the average, geometric mean, and harmonic mean. The program has a function to read the numbers and 3 different functions to obtain the average and the geometric and harmonic means.

Sample run:



#include <iostream>

using namespace std;

double average(double numbers[], int n);

double geo\_mean(double numbers[], int n);

double har\_mean(double numbers[], int n);

int main(void)

{

int n, choice;

double numbers[10];

cout << "How many numbers to process? ";

cin >> n;

cout << "Enter the numbers to process: ";

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

cin >> numbers[i];

cout << "Do you want : \n";

cout << "1. Average\n";

cout << "2. Geometric mean\n";

cout << "3. Harmonice mean\n";

cout << "Enter your choice (1-3)";

cin >> choice;

if (choice == 1)

cout << "Average is: " << average(numbers, n) << endl;

else if (choice == 2)

cout << "Geometric mean is: " << geo\_mean(numbers, n) << endl;

else

cout << "Harmonice mean is: " << har\_mean(numbers, n) << endl;

return 0;

}

double average(double numbers[], int n)

{

double total = 0.0;

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

total += numbers[i];

//average = total / n;

return total / n;

}

double geo\_mean(double numbers[], int n)

{

double mean = 0.0;

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

mean \*= numbers[i];

return pow(mean, 1.0 / n);

}

double har\_mean(double numbers[], int n)

{

double mean = 1.0;

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

mean \*= numbers[i];

return n / mean;

}