**1.**

#include <iostream>

using namespace std;

// a function to test whether a number is even

bool isEven(int x)

{

if (x % 2 == 0)

return true;

else

return false;

}

int main()

{

const int SIZE = 10;

// part a. prompt user to enter ten integer type numbers

cout << "Enter ten numbers with space in between: \n";

int number[SIZE];

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

{

cin >> number[i];

// part b. calling isEven() function and display only even integers

if (isEven(number[i]) == true)

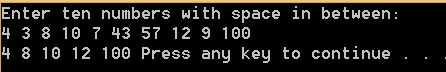
cout << number[i] << " ";

}

system("pause");

return 0;

}

****

**2.**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

// size is created as a global variable

const int SIZE = 10;

// part a. sum() function

double sum(double x[], int s)

{

double total = 0;

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

{

total += x[i];

}

return total;

}

// part b. average() function

double average(double x[], int s)

{

double ave = sum(x, s) / s;

return ave;

}

// part c. average2() function

void average2(double& num)

{

double aver = num / SIZE;

cout << "Average2 is " << aver << endl;

}

// part d. main function

int main()

{

srand(time(0));

double number[SIZE];

// assigning random integers for the array

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

{

number[i] = 1 + rand() % 100;

cout << number[i] << " ";

}

cout << endl;

// calling functions average() and print the average values

cout << "Average is " << average(number, SIZE) << endl;

// calling functions average2() and print the average values

double num2 = 0;

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

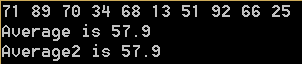
num2 = sum(number, SIZE);

average2(num2);

system("pause");

return 0;

}

****

**3.**

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

srand(time(0));

const int ROW = 4;

const int COL = 4;

int a[ROW][COL];

// writing two dimensional arrays with random numbers between 0 and 9

for (int r = 0; r < ROW; r++)

{

for (int c = 0; c < COL; c++)

{

a[r][c] = rand() % 10;

cout << a[r][c] << " ";

}

cout << endl;

}

// find smallest element in each column

for (int c = 0; c < COL; c++)

{

int small = a[0][c];

for (int r = 0; r < ROW; r++)

{

if (a[r][c] < small)

small = a[r][c];

}

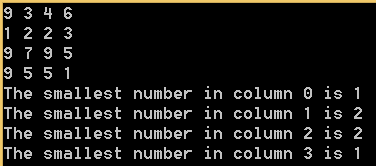
cout << "The smallest number in column " << c << " is " << small << endl;

}

system("pause");

return 0;

}

****