# Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Avoid Hard Coding of variables wherever possible
3. Proper naming conventions for all variables
4. Proper Program Name and Class Name
5. Follow proper indentation
6. Give comments for every step or logical block like a variable declaration or conditional and loop blocks
7. For every user input validate the user input, if invalid, state the error either exit the program or ask user to enter again
8. Use the Array Length property while using for loops.
9. **Sample Program 1 -** Create a program to find the sum of all the digits of a number given by a user using an array and display the sum.

**Hint =>**

1. Take the input for a number and validate, if failed state and exit the program
2. Find the count of digits in the number
3. Find the digits in the number and save them in an array
4. Find the sum of the digits of the number and display the sum

// Create SumOfDigit Class to compute the sum of all digits of a number using

// an array

using System;

class SumOfDigits

{

static void Main(string[] args)

{

// Take input for a number

Console.Write("Enter a number: ");

string input = Console.ReadLine();

if (!int.TryParse(input, out int number) || number < 0)

{

Console.WriteLine("Invalid Number.");

return;

}

// Find the count of digits

int count = input.Length;

// Extract digits and store in an array

int[] digits = new int[count];

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

{

digits[i] = number % 10;

number /= 10;

}

// Compute the sum of the digits

int sum = 0;

foreach (int digit in digits)

{

sum += digit;

}

// Display the sum

Console.WriteLine($"Sum of Digits: {sum}");

}

}



1. **Sample Program 2 -** Working with Multi-Dimensional Arrays. Write a C# program to create a 2 Dimensional (2D) array (matrix) of integers, initialize it with values, and print the sum of all elements in the matrix

**Hint =>**

1. Take the input for a number of rows and columns
2. Create a 2D array (matrix) of integers
3. Take the input for the elements of the matrix
4. Calculate the sum of all elements in the matrix and display the sum
5. Also, Display the matrix

// Program to create a 2D array, display the elements and calculate the sum of

// the elements of the array

using System;

class TwoDArray

{

static void Main(string[] args)

{

// Take input for the number of rows and columns

Console.Write("Enter the number of rows: ");

int rows = int.Parse(Console.ReadLine());

Console.Write("Enter the number of columns: ");

int cols = int.Parse(Console.ReadLine());

// Create the 2D array

int[,] matrix = new int[rows, cols];

// Input the elements of the matrix

Console.WriteLine("Enter the elements of the matrix:");

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

{

for (int j = 0; j < cols; j++)

{

Console.Write($"Element [{i},{j}]: ");

matrix[i, j] = int.Parse(Console.ReadLine());

}

}

// Display the matrix and calculate the sum

int sum = 0;

Console.WriteLine("The elements of the matrix are:");

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

{

for (int j = 0; j < cols; j++)

{

Console.Write(matrix[i, j] + " ");

sum += matrix[i, j];

}

Console.WriteLine();

}

// Display the sum of the elements

Console.WriteLine($"The sum of all elements is: {sum}");

}

}



# Level 2 Practice Programs

1. Create a program to find the bonus of 10 employees based on their years of service and the total bonus amount the company Zara has to pay, along with the old and new salary.

**Hint =>**

1. Zara decides to give a bonus of 5% to employees whose year of service is more than 5 years or 2% if less than 5 years
2. Define a double array to save salary and years of service for each of the 10 employees
3. Also define a double array to save the new salary and the bonus amount as well as variables to save the total bonus, total old salary, and new salary
4. Define a loop to take input from the user. If salary or year of service is an invalid number then ask the user to enter again. Note in this case you will have to decrement the index counter
5. Define another loop to calculate the bonus of 10 employees based on their years of service. Save the bonus in the array, compute the new salary, and save in the array. Also, the total bonus and total old and new salary can be calculated in the loop
6. Print the total bonus payout as well as the total old and new salary of all the employees

| using System;  public class Solution {  public static void Main() {  // Number of employee  int numEmployees = 10;    // declare arrays for employee detail  double[] salary = new double[numEmployees];  double[] yearsOfService = new double[numEmployees];  double[] bonus = new double[numEmployees];  double[] newSalary = new double[numEmployees];    // create variable to store old salary, new salary, bonus amount  double totalBonus = 0, totalOldSalary = 0, totalNewSalary = 0;    for (int i = 0; i < numEmployees; i++) {  // declare a variable for check valid input  bool validInput = false;    while(!validInput) {  Console.WriteLine("Enter details for Employee {0}:", i + 1);    // Get salary input  Console.Write("Enter salary: ");  salary[i] = Convert.ToDouble(Console.ReadLine());   // Get years of service input  Console.Write("Enter years of service: ");  yearsOfService[i] = Convert.ToDouble(Console.ReadLine());   // Validate Salary and years of service must be positive  if (salary[i] > 0 && yearsOfService[i] >= 0) {  validInput = true;   } else {  Console.WriteLine("Invalid input");  }  }    }    // for calculate bonus, new salary, and total amounts  for (int i = 0; i < numEmployees; i++) {  // Calculate bonus based on years of service  if (yearsOfService[i] > 5) {  bonus[i] = salary[i] \* 0.05;  } else {  bonus[i] = salary[i] \* 0.02;  }   // calculate new salary  newSalary[i] = salary[i] + bonus[i];   // update total amounts  totalBonus += bonus[i];  totalOldSalary += salary[i];  totalNewSalary += newSalary[i];  }    Console.WriteLine("Total Bonus Payout: {0}", totalBonus);  Console.WriteLine("Total Old Salary: {0}", totalOldSalary);  Console.WriteLine("Total New Salary: {0}", totalNewSalary);  } } |
| --- |

1. Create a program to find the youngest friends among 3 Amar, Akbar, and Anthony based on their ages and the tallest among the friends based on their heights

**Hint =>**

1. Take user input for age and height for the 3 friends and store it in two arrays each to store the values for age and height of the 3 friends
2. Loop through the array and find the youngest of the 3 friends and the tallest of the 3 friends
3. Finally display the youngest and tallest of the 3 friends

| using System;  class Solution {  public static void Main() {  // Declare arrays for storing the ages and heights of the 3 friends  int[] age = new int[3];  double[] height = new double[3];  string[] name = { "Amar", "Akbar", "Anthony" };   // get input for ages and heights of the three friends  Console.WriteLine("Enter the details for the three friends:");   // input for Amar  Console.Write("Enter Amar's age: ");  age[0] = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Amar's height (in cm): ");  height[0] = Convert.ToDouble(Console.ReadLine());   // input for Akbar  Console.Write("Enter Akbar's age: ");  age[1] = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Akbar's height (in cm): ");  height[1] = Convert.ToDouble(Console.ReadLine());   // input for Anthony  Console.Write("Enter Anthony's age: ");  age[2] = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Anthony's height (in cm): ");  height[2] = Convert.ToDouble(Console.ReadLine());    // declare a youngest variable  int youngestIndex = 0;  // create a loop to find youngest friend  for (int i = 1; i < 3; i++) {  if (age[i] < age[youngestIndex]) {  youngestIndex = i;  }  }    // create a variable for tallest   int tallestIndex = 0;  // loop for find tallest Index  for (int i = 1; i < 3; i++ ) {  if(height[i] > height[tallestIndex]) {  tallestIndex = i;  }  }    // print output  Console.WriteLine("{0} is youngest friend his age is {1}", name[youngestIndex], age[youngestIndex]);  Console.WriteLine("{0} is tallest friend his height is {1}", name[tallestIndex], height[tallestIndex]);    }  } |
| --- |

1. Create a program to store the digits of the number in an array and find the largest and second largest element of the array.

**Hint =>**

1. Create a number variable and take user input.
2. Define an array to store the digits. Set the size of the array to maxDigit variable initially set to 10
3. Create an integer variable index with the value 0 to reflect the array index.
4. Use a loop to iterate until the number is not equal to 0.
5. Remove the last digit from the number in each iteration and add it to the array.
6. Increment the index by 1 in each iteration and if the index count equals maxDigit then break out of the loop and the remaining digits are not added to the array
7. Define variable to store largest and second largest digit and initialize it to zero
8. Loop through the array and use conditional statements to find the largest and second largest number in the array
9. Finally display the largest and second-largest number

| using System;  class Solution {  public static void Main() {  // Declare all variable and array  int maxDigit = 10;  int []digit = new int[maxDigit];  int index = 0;    Console.Write("Enter number : ");  int number = Convert.ToInt32(Console.ReadLine());    // for storing digit in array  while(number != 0 || index < maxDigit) {  int rem = number % 10;   number /= 10;   digit[index++] = rem;   }    // declare largest and secondLargest variable  int largest = -1, secondLargest = -1;     // create loop for find largest and secondLargest value  for (int i = 0; i < index; i++) {  if(digit[i] > largest) {  secondLargest = largest;  largest = digit[i];  } else if (digit[i] > secondLargest && digit[i] < largest) {   secondLargest = digit[i];  }  }    // print largest digit  if(largest != -1) {  Console.WriteLine("Largest digit is {0}", largest);  } else {  Console.WriteLine("No largest digit found ");  }    // print second Largest digit  if (secondLargest != -1) {  Console.WriteLine("Second largest digit is {0}", secondLargest);  } else {  Console.WriteLine("No second largest digit found ");  }   }  } |
| --- |

1. Rework the program 2, especially the **Hint:** if index equals maxDigit, we break from the loop. Here we want to modify to increase the size of the array i,e maxDigit by 10 if the index is equal to maxDigit. This is done to consider all digits to find the largest and second-largest number

**Hint =>**

1. In Hint f inside the loop if the index is equal to maxDigit, increase maxDigit and make digits array to store more elements.
2. To do this, we need to create a new temp array of size maxDigit, copy from the current digits array the digits into the temp array, and assign the current digits array to the temp array
3. Now the digits array will be able to store all digits of the number in the array and then find the largest and second largest number

| using System;  class Solution {  public static void Main() {  // Declare all variable and array  int maxDigit = 10;  int []digit = new int[maxDigit];  int index = 0;    Console.Write("Enter number: ");  int number = Convert.ToInt32(Console.ReadLine());    // for storing digit in array  while(number != 0 ) {  if (index == maxDigit) {  // If the array is full, increase the size by 10  maxDigit += 10;   // Create a temporary array with the new size  int[] temp = new int[maxDigit];   // Copy existing digits into the new temp array  Array.Copy(digit, temp, digit.Length);   // Reassign the digits array to point to the new temp array  digit = temp;  }  int rem = number % 10;   number /= 10;   digit[index++] = rem;   }    // declare largest and secondLargest variable  int largest = -1, secondLargest = -1;     // create loop for find largest and secondLargest value  for (int i = 0; i < index; i++) {  if(digit[i] > largest) {  secondLargest = largest;  largest = digit[i];  } else if (digit[i] > secondLargest && digit[i] < largest) {   secondLargest = digit[i];  }  }  // print largest digit  if(largest != -1) {  Console.WriteLine("Largest digit is {0}", largest);  } else {  Console.WriteLine("No largest digit found ");  }    // print second Largest digit  if (secondLargest != -1) {  Console.WriteLine("Second largest digit is {0}", secondLargest);  } else {  Console.WriteLine("No second largest digit found ");  }   }  } |
| --- |

1. Create a program to take a number as input and reverse the number. To do this, store the digits of the number in an array and display the array in reverse order

**Hint =>**

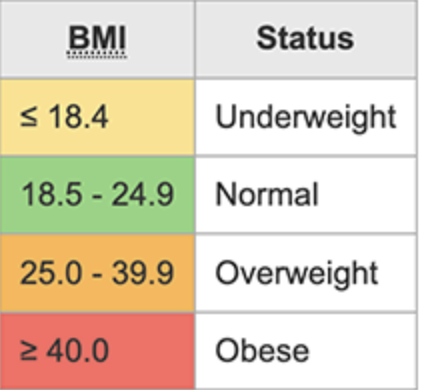
1. Take user input for a number.
2. Find the count of digits in the number.
3. Find the digits in the number and save them in an array
4. Create an array to store the elements of the digits array in reverse order
5. Finally, display the elements of the array in reverse order

| using System;  class Solution {  public static void Main() {  // take user input  Console.Write("Enter a positive number: ");  int number = Convert.ToInt32(Console.ReadLine());    // find the count of digits   int temp = number;  int digitCount = 0;   // Count the digits   while (temp != 0) {  digitCount++;  temp /= 10;  }    // Create an array to store the digits in reverse order  int[] digits = new int[digitCount];  temp = number;    // store the digits in an array  for (int i = 0; i < digitCount; i++) {  digits[i] = temp % 10;  temp /= 10;  }   Console.WriteLine("Reversed number:");  for (int i = 0; i < digits.Length; i++) {  Console.Write(digits[i]);  }  } } |
| --- |

1. An organization took up an exercise to find the Body Mass Index (BMI) of all the persons in the team. For this create a program to find the BMI and display the height, weight, BMI and status of each individual

**Hint =>**

1. Take input for a number of persons
2. Create arrays to store the weight, height, BMI, and weight status of the persons
3. Take input for the weight and height of the persons
4. Calculate the BMI of all the persons and store them in an array and also find the weight status of the persons
5. Display the height, weight, BMI, and weight status of each person
6. Use the table to determine the weight status of the person



| using System;  class Solution {  public static void Main() {  // Prompt the user for the number of persons  Console.Write("Enter the number of persons: ");  int numPersons = int.Parse(Console.ReadLine());   // Initialize arrays to store height, weight, BMI, and weight status  double[] weights = new double[numPersons];  double[] heights = new double[numPersons];  double[] bmis = new double[numPersons];  string[] weightStatuses = new string[numPersons];   // Take input for height and weight for each person  for (int i = 0; i < numPersons; i++) {  Console.Write("Enter weight of person {0} in kilograms (kg): ", i + 1);  weights[i] = Convert.ToDouble(Console.ReadLine());   Console.Write("Enter height of person {0} in centimeters (cm): ", i + 1);  heights[i] = Convert.ToDouble(Console.ReadLine());  }   // Calculate BMI and determine weight status for each person  for (int i = 0; i < numPersons; i++) {  double heightInMeters = heights[i] / 100;   bmis[i] = weights[i] / (heightInMeters \* heightInMeters);   // Determine weight status based on BMI  weightStatuses[i] = bmis[i] switch {  <= 18.4 => "Underweight",  >= 18.5 and <= 24.9 => "Normal",  >= 25 and <= 39.9 => "Overweight",  >=40 => "Obese"  };  }   // Display the results for each person  for (int i = 0; i < numPersons; i++) {  Console.WriteLine("person {0} has : height={1}, weight={2}, bmi={3}, weightStatuse={4}", i+1, heights[i], weights[i], bmis[i], weightStatuses[i]);  }  } } |
| --- |

1. Rewrite the above program using multi-dimensional array to store height, weight, and BMI in 2D array for all the persons

**Hint =>**

1. Take input for a number of persons
2. Create a multi-dimensional array to store weight, height and BMI. Also create an to store the weight status of the persons

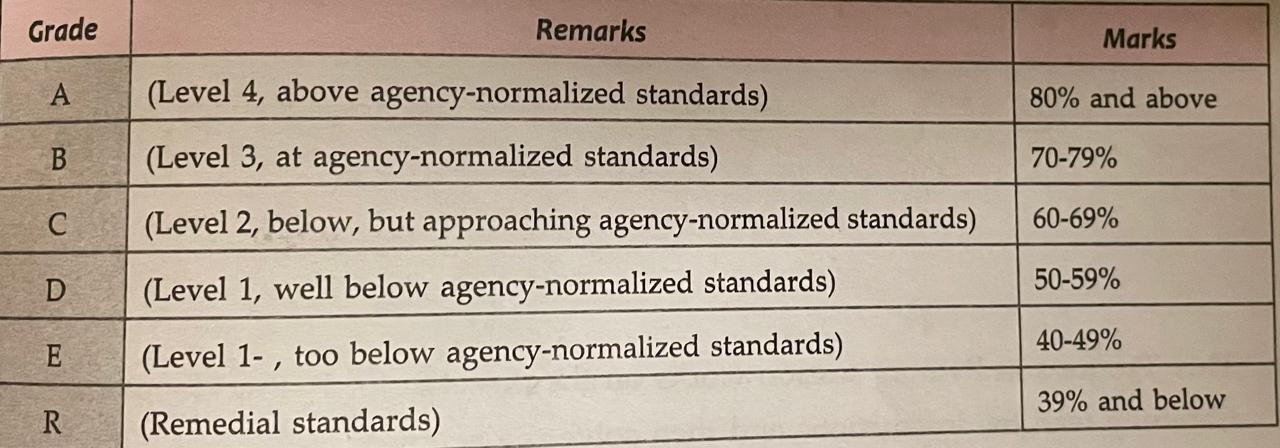
double[][] personData = new double[number][3];

String[] weightStatus = new String[number];

1. Take input for weight and height of the persons and for negative values, ask the user to enter positive values
2. Calculate BMI of all the persons and store them in the personData array and also find the weight status and put them in the weightStatus array
3. Display the height, weight, BMI and status of each person

| using System;  class Solution {  public static void Main() {  // Prompt the user for the number of persons  Console.Write("Enter the number of persons: ");  int numPersons = int.Parse(Console.ReadLine());   // Initialize a 2D array to store height, weight, and BMI  double[,] personData = new double[numPersons, 3];  string[] weightStatus = new string[numPersons];   // Take input for height and weight for each person  for (int i = 0; i < numPersons; i++) {  Console.Write("Enter height of person {0} in centimeters (cm): ", i + 1);  personData[i, 0] = Convert.ToDouble(Console.ReadLine());    Console.Write("Enter weight of person {0} in kilograms (kg): ", i + 1);  personData[i, 1] = Convert.ToDouble(Console.ReadLine());   }   // Calculate BMI and determine weight status for each person  for (int i = 0; i < numPersons; i++) {  double heightInMeters = personData[i, 0] / 100;  personData[i, 2] = personData[i, 1] / (heightInMeters \* heightInMeters);    // Determine weight status based on BMI  double bmi = personData[i, 2];  weightStatus[i] = bmi switch {  <= 18.4 => "Underweight",  >= 18.5 and <= 24.9 => "Normal",  >= 25 and <= 29.9 => "Overweight",  >=40 => "Obese"  };  }   // Display the results for each person  for (int i = 0; i < numPersons; i++) {  Console.WriteLine("person {0} has : height={1}, weight={2}, bmi={3}, weightStatuse={4}", i+1, personData[i, 0], personData[i, 1], personData[i, 2], weightStatus[i]);  }  } } |
| --- |

1. Create a program to take input marks of students in 3 subjects physics, chemistry, and maths. Compute the percentage and then calculate the grade as per the following guidelines

****

**Hint =>**

1. Take input for the number of students
2. Create arrays to store marks, percentages, and grades of the students
3. Take input for marks of students in physics, chemistry, and maths. If the marks are negative, ask the user to enter positive values and decrement the index
4. Calculate the percentage and grade of the students based on the percentage
5. Display the marks, percentages, and grades of each student

| using System;  class Solution {  public static void Main() {  // Prompt for the number of students  Console.Write("Enter the number of students: ");  int numStudents = Convert.ToInt32(Console.ReadLine());   // Initialize arrays to store marks, percentages, and grades  int[] physicsMarks = new int[numStudents];  int[] chemistryMarks = new int[numStudents];  int[] mathematicsMarks = new int[numStudents];  double[] percentages = new double[numStudents];  string[] grades = new string[numStudents];   // Input marks for each student  for (int i = 0; i < numStudents; i++) {  Console.WriteLine("Entering marks for student {0}:", i + 1);   Console.Write("Enter marks for Physics out of 100: ");  physicsMarks[i] = Convert.ToInt32(Console.ReadLine());    while (physicsMarks[i] < 0 || physicsMarks[i] > 100) {  Console.WriteLine("Marks must be between 0 and 100");  Console.Write("Enter marks for Physics out of 100: ");  physicsMarks[i] = Convert.ToInt32(Console.ReadLine());  }   Console.Write("Enter marks for Chemistry out of 100: ");  chemistryMarks[i] = Convert.ToInt32(Console.ReadLine());    while (chemistryMarks[i] < 0 || chemistryMarks[i] > 100) {  Console.WriteLine("Marks must be between 0 and 100");  Console.Write("Enter marks for Chemistry out of 100: ");  chemistryMarks[i] = Convert.ToInt32(Console.ReadLine());  }   Console.Write("Enter marks for Mathematics out of 100: ");  mathematicsMarks[i] = Convert.ToInt32(Console.ReadLine());    while (mathematicsMarks[i] < 0 || mathematicsMarks[i] > 100) {  Console.WriteLine("Marks must be between 0 and 100");  Console.Write("Enter marks for Mathematics out of 100: ");  mathematicsMarks[i] = Convert.ToInt32(Console.ReadLine());  }  }   // Calculate percentages and grades for each student  for (int i = 0; i < numStudents; i++) {  int totalMarks = physicsMarks[i] + chemistryMarks[i] + mathematicsMarks[i];  percentages[i] = totalMarks / 3.0;   grades[i] = percentages[i] switch {  >= 80 => "A",  >= 70 => "B",  >= 60 => "C",  >= 50 => "D",  >= 40 => "E",  \_ => "R"  };  }   // Display results for each student  for (int i = 0; i < numStudents; i++) {  Console.WriteLine("student {0} scored physics={1}, chemistry={2}, mathematics={3} and percentage={4} with grade={5}", i + 1, physicsMarks[i], chemistryMarks[i], mathematicsMarks[i], percentages[i], grades[i]);  }  } } |
| --- |

1. Rewrite the above program to store the marks of the students in physics, chemistry, and maths in a 2D array and then compute the percentage and grade

**Hint =>**

1. All the steps are the same as the problem 8 except the marks are stored in a 2D array
2. Use the 2D array to calculate the percentages, and grades of the students

| using System;  class Solution {  public static void Main() {  // Prompt for the number of students  Console.Write("Enter the number of students: ");  int numStudents = Convert.ToInt32(Console.ReadLine());   // Initialize arrays to store marks, percentages, and grades  int[,] marks = new int[numStudents, 3];   double[] percentages = new double[numStudents];  string[] grades = new string[numStudents];   // Input marks for each student  for (int i = 0; i < numStudents; i++) {  Console.WriteLine("Entering marks for student {0}:", i + 1);   for (int j = 0; j < 3; j++) {  string subject = j switch {  0 => "Physics",  1 => "Chemistry",  \_ => "Mathematics"  };   Console.Write("Enter marks for {0} out of 100: ", subject);  marks[i, j] = Convert.ToInt32(Console.ReadLine());   while (marks[i, j] < 0 || marks[i, j] > 100) {  Console.WriteLine("Marks must be between 0 and 100");  Console.Write("Enter marks for {0} out of 100: ", subject);  marks[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  }   // Calculate percentages and grades for each student  for (int i = 0; i < numStudents; i++) {  int totalMarks = marks[i, 0] + marks[i, 1] + marks[i, 2];  percentages[i] = totalMarks / 3.0;   grades[i] = percentages[i] switch {  >= 80 => "A",  >= 70 => "B",  >= 60 => "C",  >= 50 => "D",  >= 40 => "E",  \_ => "R"  };  }   // Display results for each student  for (int i = 0; i < numStudents; i++) {  Console.WriteLine("student {0} scored physics={1}, chemistry={2}, mathematics={3} and percentage={4} with grade={5}", i + 1, marks[i, 0], marks[i, 1], marks[i, 2], percentages[i], grades[i]);  }  } } |
| --- |

1. Create a program to take a number as input find the frequency of each digit in the number using an array and display the frequency of each digit

**Hint =>**

1. Take the input for a number
2. Find the count of digits in the number
3. Find the digits in the number and save them in an array
4. Find the frequency of each digit in the number. For this define a frequency array of size 10, Loop through the digits array, and increase the frequency of each digit
5. Display the frequency of each digit in the number

| using System;  class Solution {  public static void Main() {  // Prompt for the input number  Console.Write("Enter number: ");  int number = Convert.ToInt32(Console.ReadLine());   // Initialize the frequency array  int[] frequency = new int[10];   // Calculate the frequency of each digit  while (number > 0) {  int digit = number % 10;  frequency[digit]++;  number /= 10;  }   // printing result  for (int i = 0; i < 10; i++) {  if (frequency[i] > 0) {  Console.WriteLine("{0} occurred {1} times", i, frequency[i]);  }  }  } } |
| --- |