**Duration : 1h30 date: 21-01-2024**

**Note:**

1. Answers can be provided in either Algorithmics or C language; both are acceptable.
2. The algorithms or C programs written must include the variable declaration section.

**Exercise 1: (5 pts)**

Write an algorithm/C program that displays the biggest and the smallest divisors of a given number. (Example: biggest and smallest divisor of **10** are respectively **5** and **2**, the number itself and the number 1 are not included).

|  |  |
| --- | --- |
| **Solution** #include <stdio.h>int main() { int number, smallest, biggest; // Input the number printf("Enter a number: "); scanf("%d", &number);  smallest = number; biggest = 1; // Find the divisors for (int i = 2; i <= number / 2; i++) { if (number % i == 0) { smallest = i; biggest = number / i; break; } } if (smallest!=1){ printf("The smallest divisor of %d is: %d\n", number, smallest); printf("The biggest divisor of %d is: %d\n", number, biggest);}else printf("No divisors other than the number 1 and the number itself "); return 0;} | **(0.5pt)****(1pt)****(1pt)****(0.5)****(0.5)****(0.5)****(0.5)****(0.5)** |

**Exercise 2: (5 pts)**

- Write an algorithm/C program that:

1- Asks the user if he wants to calculate the area of a circle or a rectangle.

2- Allows the user to input data, including the width and length for a rectangle, and the radius for a circle.

3- Displays the result.

- Provide a solution using a flowchart.

|  |  |
| --- | --- |
| **Solution** #include <stdio.h>int main() { int choice; printf("Choose the shape to calculate the area:\n"); printf("1. Circle\n"); printf("2. Rectangle\n"); printf("Enter your choice (1 or 2): "); scanf("%d", &choice); if (choice == 1) { float radius, area; const PI=3.14; printf("Enter the radius of the circle: "); scanf("%f", &radius); area = PI \* radius \* radius; printf("The area of the circle with radius %.2f is: %.2f\n", radius, area); } else if (choice == 2) { double width, length, area; printf("Enter the width of the rectangle: "); scanf("%f", &width); printf("Enter the length of the rectangle: "); scanf("%f", &length); area = width \* length; printf("The area of the rectangle with width %.2f and length %.2f is: %.2f\n", width, length, area); } else { printf("Invalid choice. Please enter 1 for circle or 2 for rectangle.\n");  } return 0;} | (0.5pt)(1pt)(1pt)(0.5pt) |

(2pts)



**Exercise 3: (5 pts)**

Write an algorithm/C program that calculates the number of ‘0’ (zeros) in the lower part of a square matrix regarding its first diagonal.

**Solution**

#include <stdio.h>

int main() {

 int n=5;

 int matrix[n][n];

 int zeroCount = 0;

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

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

 if (matrix[i][j] == 0) {

 zeroCount++;

 }

 }

 }

 printf("Number of '0' in the lower part regarding the first diagonal: %d\n", zeroCount);

 return 0;

}

**Exercise 4: (5 pts)**

Write an algorithm/C program that:

1. Search for a given letter in one dimension array.
2. Move all its occurrences to the left and shift the others to the right.

#include <stdio.h>

int main() {

 int n, i, j;

 char target;

 char arr[n];

Partie declaration bien faite : 0.5

 printf("Enter the target letter to search for: "); 0.5

 scanf(" %c", &target);

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

 if (arr[i] == target) {

 // Shift elements to the right 1pt

 for (j = i; j > 0; j--) {

 arr[j] = arr[j - 1];

 } 2 pts

 arr[0] = target; 0.5

 }

 }

 return 0;

}

L’agencement 0.5

Remarque : L’enchainement est important, c’est-à-dire une instruction mal placée n’est pas comptabilisée.