**ADVANCED ALGORITHM**

*Pre-assessment Test*

|  |  |
| --- | --- |
| **Course** | Algorithm & Data Structures |
| **Exam Type** | **Pre-Assessment Test** |
| **Duration** | 2 hours |
| **Date** | SEPT 27th 2024 |
| **Exam Rules** | * Cheating with AI on a pre-assessment **doesn't help you**. * The test is to see **your real level,** not to score points. * Honest results help us to **find where you need to improve**. * **Be honest** to get the most out of the course. |

**Exercises plan**

|  |  |
| --- | --- |
| **Exercise 1** | Apply basic concepts |
| **Exercise 2** | Combine multiple concepts |
| **Exercise 3** | Use modular programming |
| **Exercise 4** | Solve complex Problems |

**Evaluation levels**

|  |  |
| --- | --- |
| **Perfect** | You solved the problem well. The code **works for all cases** |
| **Fair** | The code solves the problem correctly**. not perfect, but it mostly works** |
| **Key concepts** | You identify **key concepts** but may **have significant errors** |
| **Not the right steps** | You understand the problem but **struggles to apply the right steps** |
| **Not understood** | You **didn’t understand** the problem or made a wrong approach. |

**EXERCISE 1 (10 pts)**

We want to know if a **given number** if in a **given range.**

We process as follow:

1. Input the number, the range min and the range max
2. Output "inside" if the number is in the range [min, max], "outside" otherwise

**Input – Outputs**

|  |  |  |
| --- | --- | --- |
| **INPUT** | int | The number |
| int | The range min |
| int | The range max |
| **OUTPUT** | string | The result text |

**Examples**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **OUTPUT** | **EXPLANATION** |
| 14  10  20 | inside | *14 is in the range [10, 20]* |
| 10  10  20 | inside | *10 is in the range [10, 20]* |
| 20  18  21 | inside | *20 is in the range [18, 21]* |
| 9  10  20 | outside | *9 is out of the range [10, 20]* |
| -15  10  20 | outside | *-15 is out of the range [10, 20]* |

**Remarks**

* We assume range min is > range max

**Your code for exercise 1**

**#include <stdio.h>**

**int main()**

**{**

**// TODO**

**// Input the number, the range min and the range max**

**// Output "inside" if the number is in the range [min, max], "outside" otherwise**

**int min,max,number;**

**printf ("Input one number: ");**

**scanf ("%d" ,&number);**

**printf ("Enter the range min: ");**

**scanf ("%d" ,&min);**

**printf ("Enter the range max: ");**

**scanf ("%d" ,&max);**

**if ( number >= min && number <= max){**

**printf ("%d is inside the range [%d, %d] ", number,min,max);**

**}**

**else{**

**printf ("%d is outside the range [%d, %d] ", number,min,max);**

**}**

**return 0;**

**}**

**EXERCISE 2 (20 pts)**

We want to know among two persons **which one is the youngest!**

We process as follow:

1. Enter the first person date of birth (year/month/day)
2. Enter the second person date of birth (year/month/day)
3. Depending on the 2 dates of birth, printeither:
   * The first person is the youngest
   * The second person is the youngest
   * Both persons have the same age

**Input – Outputs**

|  |  |  |
| --- | --- | --- |
| **INPUT** | int | *person 1 year of birth* |
| int | *person 1 month of birth* |
| int | *person 1 day of birth* |
| int | *person 1 year of birth* |
| int | *person 1 month of birth* |
| int | *person 1 day of birth* |
| **OUTPUT** | string | The result text |

**Examples**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **OUTPUT** | **EXPLANATION** |
| 2002  12  31  2002  12  25 | The first person is the youngest | The first person is born in dec 31th 2002  The second person is born in dec 25th 2002 |
| 2002  12  31  2002  12  31 | Both persons have the same age | Both persons are born in dec 31th 2002 |
| 2001  12  31  2002  12  31 | The second person is the youngest | The first person is born on dec 31th 2001  The second person is born on dec 31th 2002 |

**Your code for exercise 2**

**#include <stdio.h>**

**int main()**

**{**

**// TODO**

**// Enter the first person date of birth (year/month/day)**

**// Enter the second person date of birth (year/month/day)**

**// Depending on the 2 dates of birth, print either:**

**// - The first person is the youngest**

**// - The second person is the youngest**

**// - Both persons have the same age**

**int person1\_year, person1\_month,person1\_day;**

**int person2\_year, person2\_month,person2\_day;**

**printf ("Enter the first person date of birth (year/month/day): ");**

**scanf ("%d %d %d" ,&person1\_year, &person1\_month,&person1\_day);**

**printf ("Enter the second person date of birth (year/month/day): ");**

**scanf ("%d %d %d" ,&person2\_year, &person2\_month,&person2\_day);**

**if (person1\_year > person2\_year || (person1\_year == person2\_year && person1\_month > person2\_month) || (person1\_year == person2\_year && person1\_month == person2\_month && person1\_day > person2\_day))**

**{**

**printf("\nThe first person is the youngest.\n");**

**}**

**else if (person2\_year > person1\_year || (person2\_year == person1\_year && person2\_month > person1\_month) || (person2\_year == person1\_year && person2\_month == person1\_month && person2\_day > person1\_day))**

**{**

**printf("\nThe second person is the youngest.\n");**

**}**

**else**

**{**

**printf("\nBoth persons have the same age.\n");**

**}**

**return 0;**

**}**

**EXERCISE 3 (30 pts)**

 For this exercise, we provide a start code.

* We have defined a structure Student, which contain a student grade for 5 exams.
* We have also defined an array of 5 students with their grade results

**What you need to do:**

* Complete the function calculateAverage that takes a student as an argument and returns the average grade.
* Use this function to calculate and display the average grade for each student.
* You main code shall produce the expected output

**Start code**

#include <stdio.h>

struct Student {

int id;

char name[50];

int grades[5];

};

struct Student students[5] = {

{1, "Alice", {85, 92, 78, 94, 88}},

{2, "Bob", {76, 81, 72, 85, 79}},

{3, "Charlie", {93, 89, 84, 91, 95}},

{4, "David", {65, 70, 75, 60, 80}},

{5, "Eve", {54, 68, 72, 77, 65}}

};

float calculateAverage(struct Student student) {

*// Your code here*

}

int main() {

*// Your code here*

return 0;

}

**Expected output**

Alice's average grade: 87.40

Bob's average grade: 78.60

Charlie's average grade: 90.40

David's average grade: 70.00

Eve's average grade: 67.20

**Your code for exercise 3**

**#include <stdio.h>**

**struct Student {**

**int id;**

**char name[50];**

**int grades[5];**

**};**

**struct Student students[5] = {**

**{1, "Alice", {85, 92, 78, 94, 88}},**

**{2, "Bob", {76, 81, 72, 85, 79}},**

**{3, "Charlie", {93, 89, 84, 91, 95}},**

**{4, "David", {65, 70, 75, 60, 80}},**

**{5, "Eve", {54, 68, 72, 77, 65}}**

**};**

**float calculateAverage(struct Student student) {**

**// Your code here**

**float sum = 0;**

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

**sum += student.grades[i];**

**}**

**return sum / 5;**

**}**

**int main() {**

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

**printf("%d. %s's average grade: %.2f\n", students[i].id, students[i].name, calculateAverage(students[i]));**

**}**

**return 0;**

**}**

**EXERCISE 4 (40 pts)**

For this exercise, we provide a start code:

* We define a structure **Course** containing: the course name, lecturer id, and course country
* We define a structure **Lecturer** containing: the lecturer id, first and last name
* We define list of courses and lecturers as input data

**What you need to do:**

* Display all lecturers **FIRST NAMES** who are teaching **JAVA** in **PHILIPPINES**



For this exercise, you can use the function strcmp from string.h :

strcmp(string1, string2) compares two strings character by character. If the strings are equal, it returns 0.

**Start code**

#include <stdio.h>

#include <string.h>

*// Struct for the course information*

typedef struct {

char course[20];

int lecturer\_id;

char country[30];

} Course;

*// Struct for the lecturer information*

typedef struct {

int lecturer\_id;

char first\_name[30];

char last\_name[30];

} Lecturer;

Course courses[] = {

{"java", 32, "Philippines"},

{"java", 33, "Philippines"},

{"html", 30, "Cambodia"},

{"java", 31, "Cambodia"},

};

Lecturer lecturers[] = {

{30, "Him", "Hey"},

{31, "Ronan", "Ogor"},

{32, "Gran", "Sabandal"},

{33, "Christian", "Mediola"},

};

int main() {

*// Your code here*

return 0;

}

**Output**

Gran Christian

**Explanation**

*Gran and Christian are both teaching JAVA in PHILLIPINES*

**Your code for exercise 4**