**Progress test 2**

**Student name: Huỳnh Ngọc Quang**

**Student ID: SE181838**

**Q1. Explain in your words:**

Name three types of data structures.

Answer:

* Record: a collection of related elements, possibly of different types, having a single name.
* Array: a sequenced collection of elements of the same data type.
* Linked list: a collection of data in which each element contains the location of the next element - that is, each element contains two parts: data and link

How is an element in an array different from an element in a record?

Answer:

An element in an **array** is a single value within a collection of elements of the same type, **accessed by an index**.

An element in a **record** is a field or member within a composite data structure that can hold different types of data, **accessed by its name**.

**Q2.**

Find how many times the *statement* in the following code segment in C is executed.

**Explain your answer:**

|  |
| --- |
| A = 3  while(A < 11)  **{**  statement;  A = A + 3;  **}** |

Answer:

A = 3 3 < 11 => True => Executed **statement;** A = A + 3 = 6.

A = 6 6 < 11 => True => Executed **statement;** A = A + 3 = 9.

A = 9 9 < 11 => True => Executed **statement;** A = A + 3 = 12.

A = 12 12 < 11 => False. End loop.

**Statement** is executed 3 times.

Find how many times the *statement* in the following code segment in C is executed:

|  |
| --- |
| A = 5  do  **{**  statement;  A = A + 1;  **}** while(A < 4); |

Answer:

A = 5 Execute **statement**; A = A + 1 = 5 + 1 = 6;

A < 4 => 6 < 4 => False. End loop.

**Statement** is executed 1 time.

Find how many times the *statement* in the following code segment in C is executed:

|  |
| --- |
| for (int i = 5; i < 30, i++)  {  statement;  i = i + 1;  } |

Answer:

i = 5, Execute **statement**; i = i + 1 = 5 + 1 = 6; i++ = 6++ = 7

i = 7, Execute **statement**; i = i + 1 = 7 + 1 = 8; i++ = 8++ = 9

i = 9, Execute **statement**; i = i + 1 = 9 + 1 = 10; i++ = 10++ = 11

i = 11, Execute **statement**; i = i + 1 = 11 + 1 = 12; i++ = 12++ = 13

i = 13, Execute **statement**; i = i + 1 = 13 + 1 = 14; i++ = 14++ = 15

i = 15, Execute **statement**; i = i + 1 = 15 + 1 = 16; i++ = 16++ = 17

i = 17, Execute **statement**; i = i + 1 = 17 + 1 = 18; i++ = 18++ = 19

i = 19, Execute **statement**; i = i + 1 = 19 + 1 = 20; i++ = 20++ = 21

i = 21, Execute **statement**; i = i + 1 = 21 + 1 = 22; i++ = 22++ = 23

i = 23, Execute **statement**; i = i + 1 = 23 + 1 = 24; i++ = 24++ = 25

i = 25, Execute **statement**; i = i + 1 = 25 + 1 = 26; i++ = 26++ = 27

i = 27, Execute **statement**; i = i + 1 = 27 + 1 = 28; i++ = 28++ = 29

i = 29, Execute **statement**; i = i + 1 = 29 + 1 = 30; i++ = 30++ = 31

i = 31, End loop.

**Statement** is executed 13 times.