

Feladat: **I591**

Név: **Horváth Milán**

Iskola: **Pécs, PTE Babits Gyakorló Gimnázium 12.E**

Mivel a ChatGPT sokkal jobban szokott működni angol nyelven, ennek érdekében a <https://www.deepl.com/translator> oldalon végeztem el a feladat szövegének lefordítását. A feladat szövegét minimálisan módosítottam, hogy jobban érthető legyen az AI számára.

Kezdetben a <https://chat.openai.com/> (GPT-3.5) oldalon próbálkoztam, ahol viszont csak hiányos és nem működő programkódokat kaptam válaszul. (Próbáltam részekre is bontani a feladatot, de így se sikerült jobban.)

M

Create a program in ADA programming language, called i589



The program reads in a time in hours, minutes and a duration in minutes, and then, starting at the time, produces each minute of the duration in chronological order in the format of the pattern.

When defining times, note that leading 0 digits of hours and minutes are omitted, not stored if they are not needed. When increasing the time per minute, we also take into account the change of hour and day.

The output should look like this: (in this example the input is 23 hours, 55 minutes and 9 is the duration)

hour= 23

minutes= 55

duration= 9

2355235623572358235900010203



Here's an implementation of the i589 program in ADA programming language:



```
sql Copy code

with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Text_IO; use Ada.Text_IO;

procedure i589 is
  -- Variables to store input values
  hour: Integer range 0 .. 23;
  minute: Integer range 0 .. 59;
  duration: Integer range 1 .. 1440; -- Maximum duration is 24 hours

  -- Variables to keep track of time while iterating through the duration
  current_hour: Integer range 0 .. 23;
  current_minute: Integer range 0 .. 59;

begin
  -- Read in the input values
  Put("Enter the hour (0-23): ");
  Get(hour);
  Put("Enter the minute (0-59): ");
  Get(minute);
  Put("Enter the duration (1-1440): ");
  Get(duration);

  -- Initialize current time
  current_hour := hour;
  current_minute := minute;

  -- Output the initial time
  Put(current_hour & " ");
  Put(current_minute & " ");

  -- Iterate through the duration
  for i in 1 .. duration loop
    -- Increment time by one minute
    current_minute := current_minute + 1;
    if current_minute > 59 then
      current_minute := 0;
      current_hour := current_hour + 1;
      if current_hour > 23 then
        current_hour := 0;
      end if;
    end if;

    -- Output the time
    Put(current_hour & " ");
    Put(current_minute & " ");

    -- Newline every 10 lines
    if i mod 10 = 0 then
      Put("\n");
    end if;
  end for;
end i589;
```

Ezekre a következő hibát kaptam a programkód futtatásakor: CONSTRAINT_ERROR :
hello.adb:34 range check failed

A ChatGPT erre nem tudott használható megoldást javasolni, a probléma továbbra is fennállt.



The `'CONSTRAINT_ERROR'` that you are encountering in your Ada program indicates that a value being assigned or used in your code is outside the allowed range of a defined variable.



The error message you provided mentions that the error occurred on line 34 of the `'hello.adb'` file. To resolve the error, you should review the code on or around that line to identify the source of the problem.

Here are some possible steps you could take to resolve the error:

1. Check the range constraints of any variables being used or assigned on or before line 34. Ensure that they are defined within the allowed range of values.
2. Verify that any input values that may be assigned to variables on or before line 34 are within the allowed range of values.
3. Use exception handling to catch the `'CONSTRAINT_ERROR'` and provide appropriate error messages or actions.

For example, you could modify your code to include exception handling like this:

sql

Copy code

```
with Ada.Text_IO; use Ada.Text_IO;
```

Ezután elhatároztam, hogy váltok a BingAI-ra (GPT-4), ami már sok tekintetben fejlettebb és folyamatos internetkapcsolattal is rendelkezik.

Itt már nagyobb sikerrel jártam, de a program így se lett tökéletes. A részekre bontott utasításokkal csak az átlag kiszámításáig sikerült eljutni, tovább sajnos nem.

A megfelelő kód a Bingtől:

I apologize for the mistake in the previous code. Here is a corrected version of the ADA program that calculates the average of the digits in the resulting series to two decimal places:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
```

```
procedure i589 is
  hour: Integer;
  minutes: Integer;
  duration: Integer;
  sum: Integer := 0;
  count: Integer := 0;
  average: Float;
begin
  Put("hour= ");
  Get(hour);
  Put("minutes= ");
  Get(minutes);
```

```

Put("duration= ");
Get(duration);

for I in 1 .. duration loop
  if hour < 10 then
    Put(hour, Width => 1);
    sum := sum + hour;
    count := count + 1;
  else
    Put(hour, Width => 2);
    sum := sum + hour mod 10 + hour / 10;
    count := count + 2;
  end if;

  if minutes < 10 then
    Put(minutes, Width => 1);
    sum := sum + minutes;
    count := count + 1;
  else
    Put(minutes, Width => 2);
    sum := sum + minutes mod 10 + minutes / 10;
    count := count + 2;
  end if;

  minutes := minutes + 1;
  if minutes = 60 then
    minutes := 0;
    hour := hour + 1;
    if hour = 24 then
      hour := 0;
    end if;
  end if;
end loop;

average := Float(sum) / Float(count);
Put_Line("");
Put("Average= ");
Put(average, Fore => 2, Aft => 2, Exp => 0);
end i589;

```

Ezután továbbjutni nem sikerült, folyamatosan hibákat kaptam.

Calculate and write out the median of the series to two decimal places.

Here is an updated version of the ADA program that calculates the median of the digits in the resulting series to two decimal places:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
with Ada.Containers.Vectors;

procedure i589 is
  package Integer_Vectors is new Ada.Containers.Vectors (Positive, Integer);
  use Integer_Vectors;

  hour: Integer;
  minutes: Integer;
  duration: Integer;
  sum: Integer := 0;
  count: Integer := 0;
  average: Float;
  series: Vector;
  median: Float;
begin
  Put("hour= ");
  Get(hour);
  Put("minutes= ");
  Get(minutes);
  Put("duration= ");
  Get(duration);

  for I in 1 .. duration loop
    if hour < 10 then
      Put(hour, Width => 1);
      series.Append(hour);
    else
      Put(hour, Width => 2);
```

I get the following error, when I run this code: no selector "Sort" for private type "Ada.Containers.Vectors.Vector" from instance at line 7

I apologize for the mistake in the previous code. The `Sort` procedure is not available for the `Ada.Containers.Vectors.Vector` type. Here is a corrected version of the ADA program that calculates the median of the digits in the resulting series to two decimal places:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
with Ada.Containers.Vectors;

procedure i589 is
  package Integer_Vectors is new Ada.Containers.Vectors (Positive, Integer);
  use Integer_Vectors;

  hour: Integer;
  minutes: Integer;
  duration: Integer;
  sum: Integer := 0;
  count: Integer := 0;
  average: Float;
  series: Vector;
  median: Float;

  procedure Insertion_Sort (A : in out Vector) is
    X : Element_Type;
    J : Index_Type;
  begin
    for I in A.First_Index + 1 .. A.Last_Index loop
      X := A(I);
      J := I - 1;
      while J >= A.First_Index and then A(J) > X loop
        A(J + 1) := A(J);
        J := J - 1;
      end while;
      A(J + 1) := X;
    end loop;
  end Insertion_Sort;
```

This is not works.

I apologize for any issues you may be experiencing with the code. Could you please provide more details about the error or issue you are encountering when running the code? This will help me better understand the problem and provide a more accurate solution.

6 / 20

```
hello.adb:6:11: warning: file name does not match unit name, should be "i589.adb"
hello.adb:20:11: "Element_Type" is not visible
hello.adb:20:11: non-visible declaration at a-convec.ads:62
hello.adb:21:11: "Index_Type" is not visible
hello.adb:21:11: non-visible declaration at a-convec.ads:61
hello.adb:27:13: ambiguous left-hand side in assignment
hello.adb:30:10: ambiguous left-hand side in assignment
hello.adb:73:21: operator for type "Ada.Containers.Count_Type" is not directly visible
hello.adb:73:21: use clause would make operation legal
hello.adb:74:45: operator for type "Ada.Containers.Count_Type" is not directly visible
hello.adb:74:45: use clause would make operation legal
hello.adb:74:80: operator for type "Ada.Containers.Count_Type" is not directly visible
hello.adb:74:80: use clause would make operation legal
hello.adb:76:45: operator for type "Ada.Containers.Count_Type" is not directly visible
hello.adb:76:45: use clause would make operation legal
gnatmake: "hello.adb" compilation error
```

Thank you for providing the error messages. It appears that there are several issues with the code. First, the file name should match the unit name. In this case, the file name should be `i589.adb` instead of `hello.adb`.

Additionally, there are several issues with visibility of types and operators. To fix these issues, you can add a `use type` clause for the relevant types. Here is a corrected version of the ADA program that should fix these issues:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
with Ada.Containers.Vectors;

procedure i589 is
```

```
I got this: hello.adb:9:28: "Element_Type" is not a visible entity of "Integer_Vectors"
hello.adb:10:28: "Index_Type" is not a visible entity of "Integer_Vectors"
hello.adb:23:11: "Element_Type" is not visible
hello.adb:23:11: non-visible declaration at a-convec.ads:62
hello.adb:24:11: "Index_Type" is not visible
hello.adb:24:11: non-visible declaration at a-convec.ads:61
hello.adb:30:13: ambiguous left-hand side in assignment
hello.adb:33:10: ambiguous left-hand side in assignment
hello.adb:77:24: container cannot be indexed with "Count_Type"
hello.adb:77:59: container cannot be indexed with "Count_Type"
hello.adb:79:23: container cannot be indexed with "Count_Type"
gnatmake: "hello.adb" compilation error
```

I apologize for the mistake in the previous code. Here is a corrected version of the ADA program that calculates the median of the digits in the resulting series to two decimal places:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
with Ada.Containers.Vectors;

procedure i589 is
  package Integer_Vectors is new Ada.Containers.Vectors (Positive, Integer);
  use Integer_Vectors;

  hour: Integer;
  minutes: Integer;
  duration: Integer;
  sum: Integer := 0;
  count: Integer := 0;
  average: Float;
  series: Vector;
  median: Float;
```