

Problem Set #4 - Due 03/05/03

Total 100

The purpose of this problem set is to:

- Help you become familiar with material covered in week 4 of class.

Please turn in each problem on a separate page. Each page should have your Name, email id, and the problem number clearly printed/written on it. Keep track of how long time it takes to complete each problem. The time taken for each problem should be printed on the first page. If you use more than one page for one problem, please STAPLE the pages together. You will lose points if you do not document the time taken for each problem, which at the same time means that you will get points for documenting “time taken” A template (in PDF form) is available on the web.

Problem 1 - 15 points

Part 1: What does the following code fragment do? Submit a hard copy of the output generated.

```
1. with Ada.Text_IO;
2. with Ada.Integer_Text_IO;
3. with Ada.Float_Text_IO;
4.
5. use Ada.Text_IO;
6. use Ada.Integer_Text_IO;
7. use Ada.Float_Text_IO;
8.
9. procedure Code_Fragment is
10.   A : Float ;
11.   B : Integer;
12.   C : Boolean;
13.   type Day is (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday);
14.   D : Day;
15.
16. begin
17.   B:= Integer'First;
18.   Put(B);
19.   New_Line;
20.
21.   A:= Float'Last;
22.   Put(A);
23.   New_Line;
24.
25.   C:=Boolean'Last;
26.   Put (Boolean'Image(C));
27.   New_Line;
28.
29.   D:=day'First;
30.   Put (Day'Image(D));
31.   New_Line;
```

```

32.
33. Put (Integer'Image(Day'Pos(D)));
34. New_Line;
35.
36. B:= 5;
37. Put (Day'Image(Day'Val(B)));
38. New_Line;
39.
40. Put (Day'Image(Day'Val(Day'Pos(D) -1)));
41. New_Line;
42.
43. end Code_Fragment;
44.

```

Part 2. What does the code on line 40 do? Does it raise a constraint error? Justify your answer.

Problem 2 - 20 points

Implement your version of the Type'Succ and Type'Pred functions called Successor and Predecessor that work exactly like Type'Succ and Type'Pred. The only difference being:

- Successor(Type'Last) = Type'First
- Predecessor(Type'First) = Type'Last

For example,

```
type Day is (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday);
```

```
Successor(Sunday) = Monday
```

```
Predecessor(Monday) = Sunday
```

Turn in a **hard copy** of your solution in **case study** format and the **code electronically**.

Problem 3 - 15 points

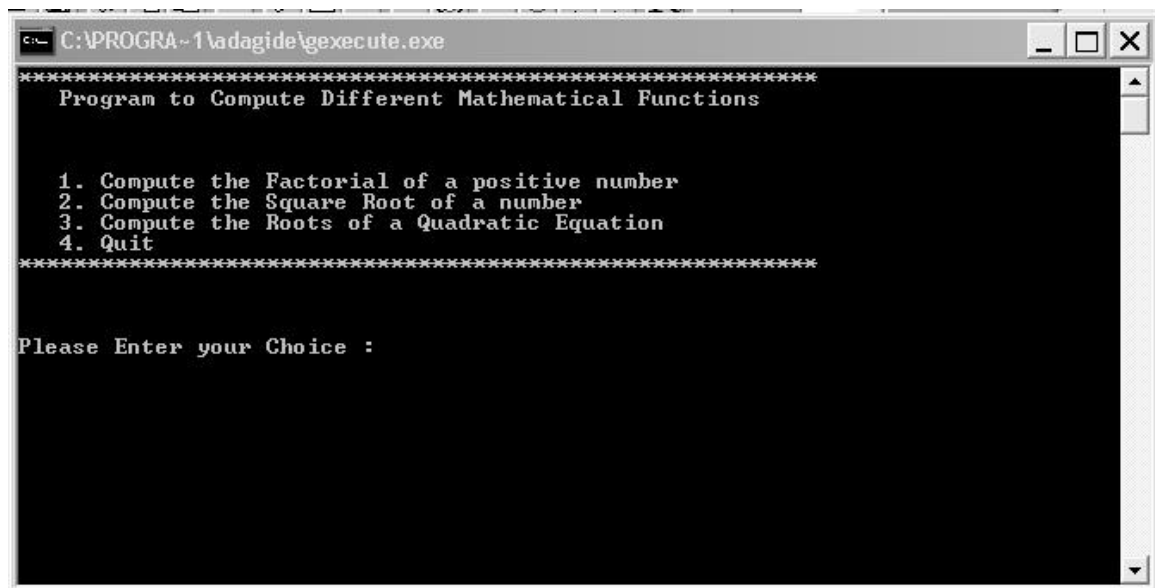
Write a program to accept a date in any one of the forms shown below.

- 21/9/2003 (Date, Month, Year)
- 21 September 2003
- 21.IX.2003 (Germany, the month is given in Roman numerals)

You may accept the day, month and year separately. Display the date accepted in all three styles. Use Enumerations to represent the month. Turn in the **hard copy** of your **algorithm** and **code listing** in class and the **code electronically**.

Problem 4 - 50 points

Write a menu-driven program that computes the different mathematical functions. The main menu is shown below.



```
C:\PROGRA~1\adagide\gexecute.exe
*****
Program to Compute Different Mathematical Functions
*****
1. Compute the Factorial of a positive number
2. Compute the Square Root of a number
3. Compute the Roots of a Quadratic Equation
4. Quit
*****
Please Enter your Choice :
```

Use separate files to implement each of the menu choices.

1. When the user selects 1,2 or 3,
 - a. Clear the screen
 - b. Prompt the user for the required inputs. Your program should be robust (i.e. if the user inputs the wrong/ out of range numbers, it should inform him/her that the input was wrong and skip to step e. **Do not** use exceptions for robustness.)
 - c. Compute the value
 - d. Display the result.
 - e. Prompt the user “Continue (Y/N) ”.
 - i. If the user selects N, quit the program.
 - ii. If the user selects Y, clear the screen and display the main menu shown above.
2. If the user selects 4,
 - a. Prompt the “user Sure you want to quit” (Y/N).
 - i. If the user selects Y, quit the program.
 - ii. If the user selects N, clear the screen and display the main menu shown above.

Turn in a **hard copy** of your solution in **case study format** and an **electronic copy** of your **code**.

You may use existing math library routines.

Hint :

1. Use Enumerations for the choices.
2. Use a case statement for the main menu
3. What loops do you need to implement the program?