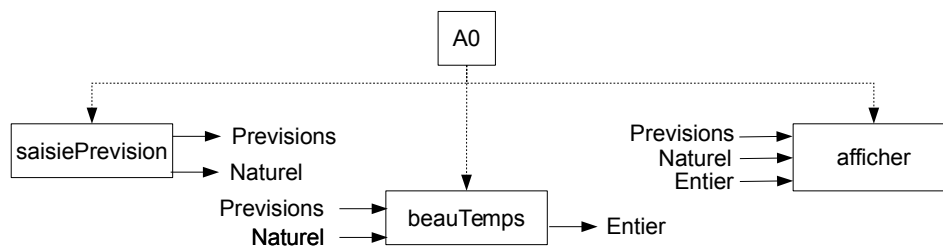


1 Prévisions météorologiques

Vous devez écrire un programme qui permet de saisir une suite de prévisions météo, de calculer les jours de soleil prévus puis de les afficher à l'écran.

Votre programme doit respecter l'analyse descendante donnée ci-dessous.



Avec pour types de données définis :

- Type temps = (orage, pluie, averse, éclaircies, soleil); qui représente les modalités de prévisions météo possibles, et
- Type Previsions = array[1..MAX] of temps; qui représente les prévisions successives des jours suivants.

Et pour signature des sous-programmes :

- procédure saisiePrevision(var meteo : Previsions; var nb : Integer) qui demande à l'utilisateur de saisir le nombre de prévisions qu'il souhaite entrer (vérifier que ce nombre est inférieur à MAX), puis les différentes valeurs. À l'issue de l'exécution de cette procédure, les paramètres de sortie ont les valeurs saisies. On considère que l'utilisateur n'entre que des modalités valides.
- fonction beauTemps(meteo : Previsions; nb : integer) : Integer qui calcule le nombre de jours prévus où le soleil apparaît (*i.e.* la prévision a pour valeur éclaircies¹ ou soleil).
- procédure afficher(meteo : Previsions; nb : integer; joursBeauTemps : Integer) qui affiche à l'écran le nombre de jour de beau temps, puis les indices des jours ayant pour modalité "soleil".

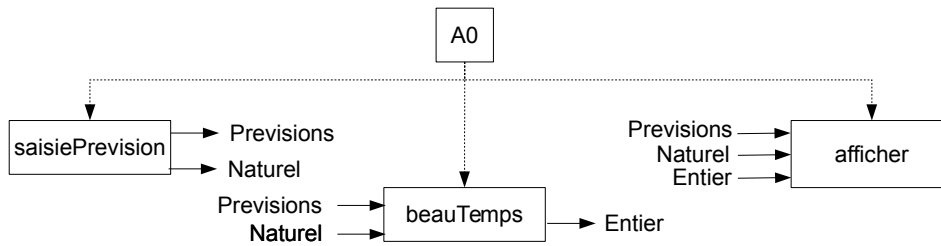
Le programme principal ne fera qu'appeler les procédures et fonctions.

2 Weather Forecasts

You must write a program that allows you to enter a series of weather forecasts, calculate the expected sunny days and then display them on the screen.

1. Nous sommes en Normandie, l'optimisme est nécessaire.

Your program must respect the given top-down analysis below.



With defined data types :

- Type `Weather` = (storm, rain, shower, cloudy, sun); who represents the possible weather forecasting modalities, and
- Type `Forecasts` = `array[1..MAX]` of time; which represents the successive forecasts for the following days.

And for signing the subprograms :

- `procedure` `entryPrevision`(var `meteo` : `Forecasts`; var `nb` : `Integer`) which asks the user to enter the number of forecasts he wants to enter (check that this number is less than `MAX`), then the different values. At the end of this procedure, the output parameters have the values entered. The user is considered to enter only valid terms and conditions.
- `function` `beauTemps`(`meteo` : `Forecasts`; `nb` : `integer`) : `Integer` which calculates the number of days the sun appears (*i.e.* the forecast has the value cloudy² or sun).
- `procedure` `display`(`meteo` : `Forecasts`; `nb` : `integer`; `daysBeauTemps` : `Integer`) which displays on the screen the number of days of good weather, then the indices of days with the mode "sun".

The main program will only call up the procedures and functions.

3 Populations

Écrire un programme qui permet de saisir les populations par année d'un pays, d'en afficher une analyse puis de les enregistrer dans un fichier. On considérera seulement 10 années consécutives.

Exercices

1. Définissez par un type de données un tableau pour représenter ces valeurs.
2. Écrire une procédure pour remplir toutes les valeurs d'un tableau de ce type. Cette procédure est appelée en première instruction du programme principal.
3. Écrire une procédure, appelée en seconde instruction du programme principal qui affiche à l'écran une analyse des chiffres saisis. Cette analyse doit faire apparaître :
 - l'année où la population la plus faible.

2. We are in Normandy, optimism is necessary.

- L'évolution de la population (c-à-d l'augmentation ou la baisse de population, avec leur valeur) pour chaque pays par année.
4. Écrire une troisième procédure appelée avant que le programme se termine qui enregistre l'ensemble des populations saisies dans un fichier `population.txt`
-

4 Populations

Write a program that captures the populations by year of a country, displays an analysis and stores them in a file. Only 10 consecutive years will be considered.

Exercices

1. Define by a data type an array to represent these values.
 2. Write a procedure to fill in all the values of a table of this type. This procedure is called in first instruction of the main program.
 3. Write a procedure, called in the second instruction of the main program that displays an analysis of the numbers. This analysis must reveal :
 - the year with the lowest population.
 - The evolution of the population (i.e. the increase or decrease of population, with their value) for each country per year.
 4. Write a third procedure called before the program ends, which records all populations entered in a file `population.txt`
-