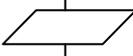
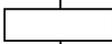
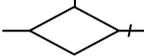


Notion d'organigramme

Avant de commencer la traduction d'un algorithme pour produire un programme, il est préférable de vérifier le raisonnement exprimé par l'algorithme. Pour ce faire, on utilise souvent un outil qu'on appelle « organigramme ». Simplement, un organigramme est une représentation graphique d'un algorithme utilisant des symboles normalisés. Plus précisément, c'est une structure constituée d'une suite de directives, composées d'actions et de décisions, qui doivent être exécutées selon un enchaînement strict pour réaliser une tâche (ou séquence).

Quelques symboles utilisés pour construire un organigramme

On donne dans le tableau ci-dessous, quelques symboles qui sont fréquemment utilisés pour la construction des organigrammes.

| Symbole | Description |
|---|--|
|  | Début ou fin d'un algorithme. |
|  | Mise à disposition ou enregistrement d'une information (lire ou écrire). |
|  | Opération sur des données, instructions, etc. |
|  | Décision d'un choix parmi d'autres en fonction des conditions. |
|  | Appel d'un sous-programme. |
|  | Pointe la séquence d'instruction suivante. |

Exemple 1

Ecrire un algorithme qui permet de calculer la surface d'un rectangle. Réaliser l'organigramme correspondant à l'algorithme.

Algorithme Surface_rectangle ;

Var

long, larg : réel ;

Surface : réel ;

Début

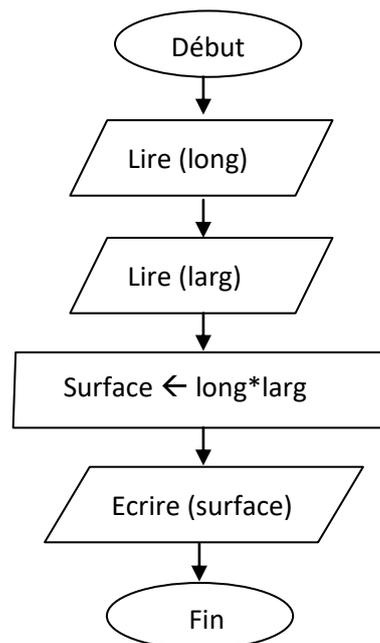
Lire (long) ;

Lire (larg) ;

Surface \leftarrow long*larg ;

Ecrire (surface) ;

Fin.



Exemple 2

Ecrire un algorithme qui vérifie si un nombre saisi par le clavier est pair ou impair. Réaliser l'organigramme correspondant à l'algorithme.

Algorithme pair_impair ;

Var

 nbr : entier ;

Début

 Lire (nbr) ;

Si (nbr mod 2 = 0) **alors**

 Ecrire ('Nombre pair') ;

Sinon

 Ecrire ('Nombre impair') ;

Finsi ;

Fin.

