

TME3 - ISS

Exercice 1.

1.1 et 1.2

```
int main(int argc, char* argv[]){
    if (argc <= 1){
        fprintf(stdout, "Usage: %s <liste_nom>", argv[0]);
    }

    int idx = 1;

    while (idx < argc){
        printf("%s\n", argv[idx]);
        idx++;
    }

    return 0;
}
```

1.3

On doit demander à l'utilisateur d'insérer à chaque fois et sauvegarder dans un array.

(à vérifier)

Exercice 2.

2.1

```
int main(int argc, char* argv[]){
    if (argc <= 1){
        fprintf(stdout, "Usage: %s <liste_nom>", argv[0]);
    }

    int idx = 1;
    float res = 0;

    while (idx < argc){
        float += atof(argv[idx]);
        idx++;
    }

    printf("%f", res);

    return 0
}
```

2.2

```
$ type true # true is a shell builtin
$ true
$ echo $? # 0
```

```
$ type false # false is a shell builtin
$ false
$ echo $? # 1
```

2.3

```
#!/bin/bash

somme=0

while true ; do
    echo "Saisissez un nombre ou 'q' pour quitter : " ; read lect

    if [[ $lect == 'q' ]] ; then
        break
    fi

    echo "Vous avez ajouté $somme à la somme"

    # On assume que si non est q alors est un nombre.
    somme=$((somme + lect))
done
```

2.4

Le même nombre que de tours de boucle.

2.5

Exercice 3.

3.1

```
#include <stdlib.h>
#include <stdio.h>

char *temp = NULL;

char var[3][5] = {"HOME", "PATH", "PWD"};
int x;

for (x=0; x < 3; x++){
    temp = getenv(var[x]);
    if (temp != NULL){
        printf("%s", temp);
    } else {
        printf("%s is null", var[x]);
    }
}
```

3.2

Que dans le cas d'export, la variable 'survive' et est dans tous les subprocessus exécutés par le shell, ce qui n'est pas le cas dans `TEST_2="J'existe"` monEnvironnement qui est équivalent à faire un `env` et donc seulement pour cette exécution là.

3.3

Sur machine.

3.4

Comme `env` est une commande externe, on peut l'utiliser dans n'importe quel lieu sans dépendre de l'état interne de la session du shell.

3.5

Fait sur machine.

3.6

`unset` est une commande intégrée et c'est normal car elle modifie l'environnement du shell. Sans être une commande intégrée, elle n'aurait pas le droit de modifier le terminal parent (car cela créerait un nouveau processus).

