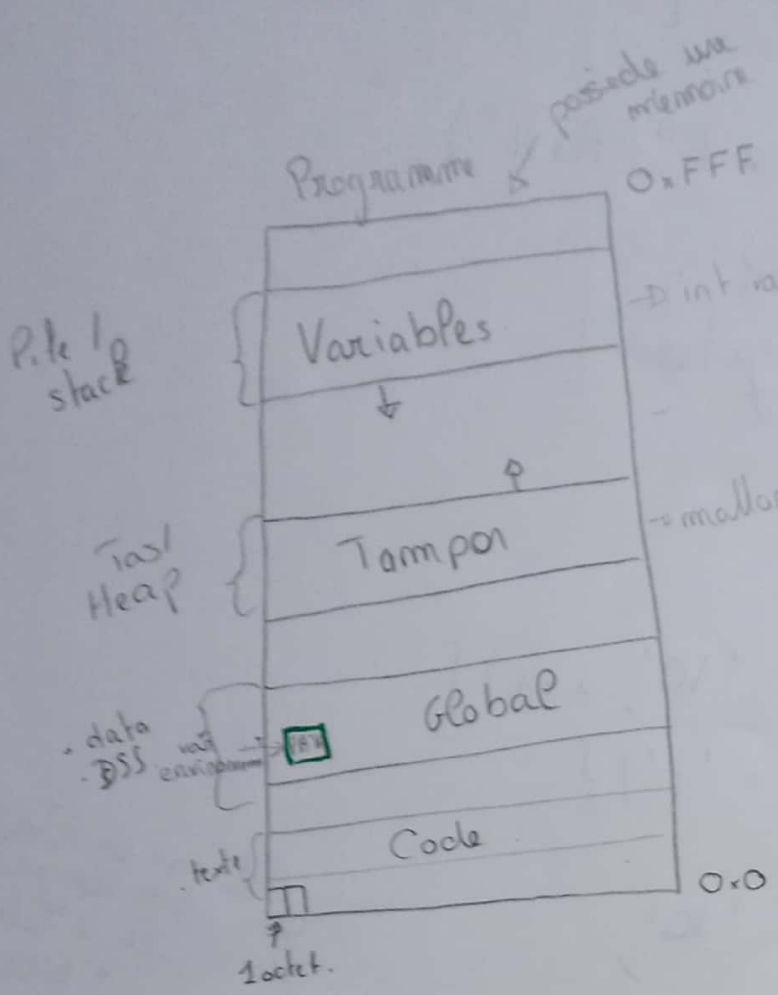
TD2 : PDS :



⚠ si on modifie l'environ

getenv("...");

↑

lire une variable d'environnement

```
char ** dirs;                                   variable global
void filldirs(void) {
    int n;
    int i, j;
    char * p = getenv("PATH");
    if (p == NULL) {
        perror("getenv");
        exit(1);
    }
    while
    for(i=0, p[i] != '\0', i++) {
        if (p[i] == ':') {
            n++;
            p[i] = '\0';
        }
    }
    dirs = malloc(sizeof(char) * (n+2));
    for(i=0, j=0; i < n; j++) {
        if (p[j] == '\0') {
            i++;
            dirs[i] = &p[j+1];
        }
        dirs[i] = NULL;
    }
}
```

+2 car on n'a pas compté la 1ère chaine + NULL

Q2. #include <limits.h>

```
int which(char * cmd) {
    char * s [PATH_MAX+1];
    for (int i=0; dirs[i] != NULL; i++) {
        snprintf(s, PATH_MAX, "%s/%s", dirs[i], cmd);
        if
        int res = access(~~cmd~~ s, X_OK);
        assert( r != -1);
        if (r == 1) {
            return 1;
        }
        else return -1;
    }
```

Q3.

```
int main(int argc, char * argv[]) {
    filldirs();
    for (int i=1; i < argc; i++) {
        if (which(~~arg~~ argv[i]) == -1) {
            return 1; ← mal fonctionné
        }                ou ; ← permet de savoir quel commande
        else                  est mal passé
            return 0. ← tout se passe bien
    }
```