

Travaux dirigés n°14 (2h15):

Portée des variables et allocation dynamique

Objectifs

- Comprendre la notion de portée (locale, globale).
- Comprendre comment déclarer des variables dynamiques
- Gérer l'allocation et la désallocation de la mémoire.

Exercice 1: Portée des variables.

Soit le programme suivant (qui ne fait rien de spécial), donner les affichages réalisés par les différents *printf*.

```
#include<stdio.h>
int i=100, j=10;

void bidon(int i)
{
    int j=20;
    printf("%d,%d",i,j);
}
void rebidon(int* k)
{
    *k = 25;
    j = j + 20;
}

void rerebidon(int i, int j)
{
    i=i+10;
    j=j+90;
    printf("%d,%d",i,j);
}

int main(void)
{
    printf("%d,%d",i,j);
}
```

```
bidon(56);
rebidon(&i);
printf("%d, %d", i,j);
rerebidon(i,j);
printf("%d, %d", i,j);
}
```

Exercice 2: Utilisation simple de malloc et free

- a) Écrire un programme en C qui alloue dynamiquement la place pour un entier qui demande une valeur à l'utilisateur pour cet entier et qui l'affiche à l'écran avant de désallouer l'espace réserver pour cet entier.
- b) Écrire un programme qui demande à l'utilisateur la taille d'un tableau d'entiers, qui réserve la place exacte pour ce tableau, qui demande les valeurs successives de ce tableau à l'utilisateur puis qui affiche le tableau et enfin la place prise par le tableau et qui répète ces opérations 5 fois.

Exercice 3: structures et allocation dynamique

Reprendre le TD sur les polygones. Dans cet exercice le tableau de points devra avoir la taille exacte du nombre de points qui constitue le polygone en utilisant l'allocation dynamique.

- a) Proposez une nouvelle structure *polygone*.
 - b) Comment déclarer un tableau de polygones dont la taille sera donnée par l'utilisateur (donc allocation en espace dynamique).
 - c) Écrire en C la fonction *creerPoly* qui permet de créer (**en allocation dynamique**) un nouveau polygone. La fonction a en paramètre le nombre de points du polygone. La fonction demande à l'utilisateur les points qui constituent ce nouveau polygone et retourne un pointeur sur ce polygone créé.
 - d) Écrire en C un programme principal qui demande la taille du tableau de polygones à l'utilisateur et qui crée le tableau de polygones (en allocation dynamique) correspondant (en créant chacun des polygones grâce à la fonction *creerPoly* de c)).
 - e) Écrire une procédure en C *supprPoly* qui reçoit un polygone de la forme définie en a) et qui désalloue la place occupée par ce polygone.
-