

# DS: Programmation C

29 Janvier 2021 - 14h00/15h30

Nom: ..... Prénom: .....

Documents autorisés : une feuille A4 recto/verso manuscrite.

## 1 Chaîne de compilation

**Question 1** À quoi servent les options de compilation `-c` et `-lm` ?

**Question 2** Listez toutes les phases de la compilation pour un projet qui contient fichier1 .c, fichier2 .c, main.c (fichier contenant la fonction principale).

## 2 Types entiers

**Question 3** Quelles lignes de code vous paraissent douteuses (en supposant qu'elles sont mises bout à bout). Justifiez à côté.

- `char x = 'e';`
- `int i = x - 200;`
- `printf ("%x\n", 0x53835);`
- `printf ("%c\n",x);`

**Question 4** Écrivez la fonction `long set_bit(long bitfield , int numbit)` qui positionne le numbit-ième bit de `bitfield` à 1.

### 3 Tableaux et pointeurs

**Question 5** Considérez le programme suivant :

```
int tab[] = {3,5,7,9,11,13,15,17,19};
int *p;
p = tab;
```

Quelles valeurs ou adresses portent les expressions suivantes (pour les adresses, indiquez le nombre **d'octets** de décalage par rapport à une adresse connue) :

1. (\*p)+2
2. tab+3
3. \*(p+2)
4. &tab[4]-p
5. p+3
6. p+(\*p+4)
7. tab[5]-3

**Question 6** Soit le programme suivant s'exécutant sur une machine 64 bits :

```
main ()
{
    int A = 2;
    int B = 4;
    int C = 8;
    int *P1, *P2, *P3;
    P1=&A;
    P2=&C;
    P3 = P1++;
    *P1=(*P2)++;
    P1=P2;
    P2=&B;
    *P1--=*P3;
    ++*P2;
    *P3*=*P2;
    A=++*P2**P1;
    P1=&A;
    *P2=*P1/=*P2;
    return 0;
}
```

Complétez le tableau suivant qui contient le contenu de chaque variable après chaque instruction, en considérant que les variables A, B, C, P1, P2 et P3 sont stockées de manière consécutive depuis l'adresse 0xff8340 (pour A) vers des adresses croissantes :

	A	B	C	P1	P2	P3
(déclarations)	2	4	8			
P1=&A	1	2	3	0xff8340		
P2=&C						
P3=P1++						
*P1 = (*P2)++						
P1=P2						
P2=&B						
*P1 -= *P3						
++*P2						
*P3* = *P2						
A = ++*P2**P1						
P1=&A						
*P2 = *P1 / = *P2						

**Question 7** Pourquoi la fonction suivante n'a-t-elle pas le comportement attendu ?

```

int* create_array(int init_value)
{
    int i = 0;
    int new_array[20];
    for (i = 0; i < 20; i++) {
        new_array[i] = init_value;
    }
    return new_array;
}

```

**Question 8** Proposez une manière d'implanter create\_array de la question précédente qui soit correcte.

**Question 9** Quel est le résultat d'un sizeof sur chacune de ces variables locales ?

```

char c = 0x12;
int i = 42;
int itab[6] = {0, 11, 22, 33, 44, 55};

```

```
int* p1 = &tab;
char* ctab = "chaine";
```

**Question 10** Écrivez une fonction qui affiche, sur une ligne différente, chaque octet d'un entier non signé passé en paramètre. Faites en sorte que votre fonction soit correcte quelle que soit la taille des entiers non signés (qui dépend du système rappelons le).

## 4 Chaînes de caractères

**Question 11** Définissez grâce à une directive du préprocesseur un entier représentant la taille maximale d'un mot français ; on considère que c'est le mot *hippopotomonstrosesquipédaliophobique* (qui signifie la peur de prononcer des mots trop longs) qui est maintenant le plus long de la langue française.

**Question 12** Définissez un type tableau `mot_fr_t` pour un mot français.

**Question 13** Écrire une fonction `bool est_facteur_gauche(mot_fr_t facteur_g, mot_fr_t src)` qui vérifie si `facteur_g` est un facteur gauche de `src`. Par exemple, "diction" est un facteur gauche de "dictionnaire"

**Question 14** Quelle taille en octets prennent chacun des paramètres `facteur_g` et `src` ?

## 5 Gestion de fichiers et `malloc()`

Voici le prototype des fonctions que vous pouvez utiliser pour manipuler les fichiers :

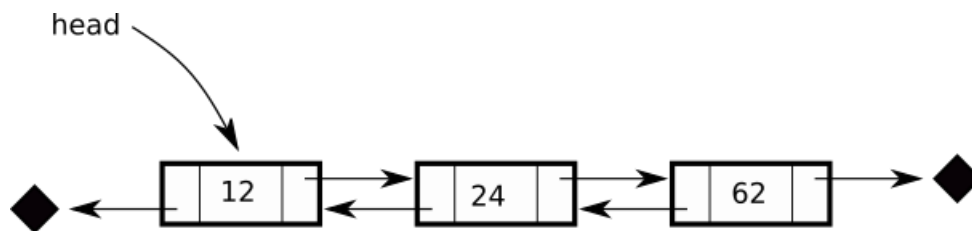
- `int feof(FILE* f)`; retourne EOF si la fin d'un fichier est atteinte
- `FILE *fopen(const char *pathname, const char *mode)`;
- `int fclose(FILE *stream)`;
- `int fgetc(FILE *stream)`; renvoie le prochain `char` lu dans `stream` ou EOF.
- `char *fgets(char *s, int size, FILE *stream)`; lit au plus `size - 1 char` dans `stream` ; s'arrête de lire quand une nouvelle ligne est atteinte, ou la fin du fichier.
- `int fputc(int c, FILE *stream)`; écrit le caractère `c` dans `stream`
- `int fputs(const char *s, FILE *stream)`; écrit la chaîne `s` dans `stream`, sans le caractère null de fin.

**Question 15** Écrire une fonction `int concat(char* nomFichier, char* nomFichier1, char* nomFichier2)` concaténant les fichiers de nom `nomFichier1` et `nomFichier2` en un fichier `nomFichier`. La fonction doit retourner 0 s'il y a eu une erreur (par exemple les noms des fichiers ne sont pas corrects) et un autre entier si tout s'est bien passé.

**Question 16** Écrire une fonction `char* read_nieme_ligne(int n, char* nom)` qui renvoie la n-ième ligne du fichier dont le nom est passé en paramètre.

## 6 Problème : liste doublement chaînée triées.


On considère une structure de liste doublement chaînée. Pour représenter de telles listes des cellules contenant un entier et deux pointeurs : un vers la cellule précédente et un vers la cellule suivante. La cellule de tête n'a pas de cellule précédente (donc pointe vers NULL) et la cellule de fin n'a pas de cellule suivante (donc pointe vers NULL également). Graphiquement la liste contenant les valeurs successives 12, 24, 62, et pointée par head est représentée de la façon suivante (le pointeur NULL est représenté par un losange noir) :



**Question 17** Donnez l'implantation d'un type `dbl_list` permettant d'implanter une telle structure.

**Question 18** Etant donnée une variable `ptr` de type `dbl_list *`, donnez la suite d'instructions pour implanter la liste présentée de manière schématique ci-avant.

**Question 19** Ecrire une fonction insertion permettant d'insérer un élément dans une `dbl_list` en considérant que la liste passée en argument est ordonnée par ordre croissant, et de telle manière à ce que la liste reste triée.



**Question 20** Utilisez le cadre suivant pour insérer toutes les indications utiles que vous n'avez pas eu la place de mettre avant. Notez bien le numéro de la question suivi du commentaire que vous jugez bon d'ajouter.

