



DEV3 - CPPL

Interrogation n° 1

R. Absil (abs)
Vendredi 27 novembre

/28

Consignes générales

1. L'interrogation dure 2h.
2. L'interrogation est à cahier ouvert : vous disposez de toutes les ressources *en lecture* que vous souhaitez.
3. Vous devez remettre vos travaux via git, en créant vous-même un repository, ajouter M. Absil au rôle de maintenir, et lui communiquer l'url par email.
4. Vous devez effectuer un commit *et* un push à chaque fois que vous passez d'une (sous-)question à une autre, p. ex. si vous passez à la question 3 après la 1.4 en sautant la 2. Les remises ne respecteront pas cette consigne *ne* seront *pas* corrigées
5. Tout code qui ne compile pas est compté comme nul.
6. Les lignes de code induisant des fuites mémoires sont sanctionnées à hauteur de 5 points.

Question 1. Créez soit

- un projet C11 au sein de Qt Creator,
- un projet C11 ne nécessitant pas d'IDE, en fournissant un Makefile, et un readme détaillant la target à utiliser pour compiler, et la commande à taper pour lancer votre projet.

/1

Question 2. Importez au sein de votre projet toutes les fonctions métiers de vos laboratoires C11 relatives aux nombres premiers.

/1

Question 3. Définissez dans `primestat.h` et `primestat.c` une fonction `int extr_decomp(int lower, int higher, bool mult)` qui, dans l'intervalle $[lower, higher]$, retourne l'entier qui a un nombre de facteurs premiers maximum,

/8

- en tenant compte des multiplicités si `mult` est `true`,
- en n'en tenant pas compte sinon.

Vous allez peut-être être amenés à écrire des fonctions helpers pour cette définition. Il est attendu que vous limitiez le copier / coller dans leur implémentation.

L'extrait de code ci-dessous illustre le fonctionnement de cette fonction.

```
1 //                                mult
2 int n1 = extr_decomp(7, 10, true); //n1 = 8 car 3 fac. pr.
3 int n3 = extr_decomp(7, 10, false); //n3 = 10 car 2 fac. pr. !=
```

Question 4. Dans `main.c`, calculez

/2

- `extr_decomp(5000, 20000, true);`
- `extr_decomp(5000, 20000, false);`

Question 5. Définissez dans un fichier `statsample.h` la structure `StatSample` suivante :

/1

```
1 struct StatSample
2 {
3     int count;
4     int sum;
5     int* data;
6 };
```

Définissez également un alias de type pour que `StatSample` dénote `struct StatSample`.

Le but de cette structure est de maintenir des informations statistiques à propos d'un échantillon de données entières, à savoir le nombre de données maintenues, leur somme, et l'adresse vers un espace mémoire contenant l'échantillon en question.

Question 6. Définissez dans `statsample.h` et `statsample.c` une fonction `void init_stat(StatSample* s)` initialisant à zéro les attributs `count` et `sum` d'un `StatSample` passée par adresse, et plaçant dans `data` l'adresse d'un espace dynamiquement alloué permettant d'y stocker 10 entiers.

/1

Question 7. Définissez dans `statsample.h` et `statsample.c` une fonction `void free_stat(StatSample* s)` libérant l'espace dynamiquement alloué dénoté par l'attribut `data` non `NULL` de `s`.

/1

Question 8. Définissez dans `statsample.h` et `statsample.c` une fonction `void update_stat(int n, StatSample* s)`, ajoutant l'entier `n` à un échantillon, en mettant à jour les attributs `count` et `sum` de la structure `UStat` associée.

/6

Si il n'y a plus de place dans l'espace dénoté par `data` pour stocker `n`, un espace de taille deux fois plus grand est alloué, le contenu de l'espace dénoté par `data` y est recopié et `data` est mis à jour pour dénoter ce nouvel espace.

L'extrait de code ci-dessous illustre le fonctionnement de cette fonction.

```

1 StatSample s;
2 init_stat(&s);           //count = 0    sum = 0
3 for(int i = 1; i <= 10; i++)
4     update_stat(i, &s);
5 //après la boucle, count = 10    sum = 55
6
7 update_stat(42, &s); //l'espace dénoté par data soit être réalloué

```

Question 9. Définissez dans `statsample.h` et `statsample.c` une fonction `int sample_avg(StatSample s)` calculant la moyenne de l'échantillon `s`. Cette fonction doit s'exécuter en temps constant.

/1

Question 10. Dans `main.c`, construisez un échantillon statistique avec le nombre de facteurs premiers des nombres de l'intervalle `[4 000, 20 000]`, en tenant compte des multiplicités. Calculez la moyenne de cet échantillon.

/6