Organisation des ordinateurs
Examen de juin 2021
Énoncés et solutions

Note : En raison des mesures sanitaires qui étaient en vigueur en 2021, cet examen était plus court qu'habituellement.

Énoncés

1. Dans le système de géolocalisation GPS, une position prend la forme

$$lat_d\,^\circ\, lat_m\,'\, lat_s\,'' \, H \;\; long_d\,^\circ\, long_m\,'\, long_s\,'' L,$$

où
— lat_d, $long_d$, lat_m, $long_m$, lat_s et $long_s$ sont des nombres entiers.
— lat_d, $long_d \in [0, 90]$.
— lat_m, $long_m$, lat_s, $long_s \in [0, 59]$.
— H est égal à N ou S.
— L est égal à E ou W.

   (a) Si toutes les positions sont équiprobables, quelle quantité d'information une position contient-elle ?

   (b) Un bateau navigue entre les 30ème et 50ème parallèles de latitude sud, en d'autres termes, sa position est telle que

$$30\,^\circ\,0\,'\,0\,'' \;\leq\; lat_d\,^\circ\, lat_m\,'\, lat_s\,'' \leq 50\,^\circ\,0\,'\,0\,'' \text{ et } H = \text{S}.$$

   Si toutes les positions satisfaisant ces contraintes sont équiprobables, quelle est dans ce cas particulier la quantité d'information fournie par une position du bateau ?

2. (a) Quel est le plus petit nombre de bits permettant de représenter le nombre -2021 par valeur signée, par complément à un et par complément à deux ? (Justifier votre réponse.)

   (b) Calculer 17,625 − 21,5 en représentant les nombres en virgule fixe avec 8 chiffres avant et 4 chiffres après la virgule.

   (c) Calculer la représentation du nombre $-3 \,.\, 2^{42}$ par le procédé IEEE754 en simple précision.

(d) Existe-t-il des nombres entiers négatifs possédant la même représentation par complément à un et par complément à deux ? (Justifier votre réponse.)

3. Qu'est-ce que le compteur de programme du processeur ? Que contient-il quand le processeur exécute une instruction donnée d'un programme ?

4. On souhaite programmer une fonction prenant comme arguments l'adresse `p` d'un tableau contenant des entiers signés représentés sur 16 bits, et le nombre `n` d'éléments de ce tableau représenté de façon non signée sur 32 bits. On suppose que l'on a $\texttt{n} \geq 1$.

   La fonction doit retourner le nombre de valeurs identiques consécutives situées à la fin du tableau. Par exemple, pour les tableaux $[-2, -2, 2, 5, 2]$ et $[0]$, la fonction doit retourner 1. Pour le tableau $[-7, 2, -7, -7, -7]$, elle doit retourner 3.

   (a) Écrire, en pseudocode ou en langage C (au choix), un algorithme permettant de résoudre ce problème.

   (b) Traduire cet algorithme en assembleur x86-64, en veillant à respecter la convention d'appel de fonctions des systèmes *Unix*.

Exemples de solutions

1. (a) Les valeurs de lat_d, $long_d$, lat_m, $long_m$, lat_s, $long_s$, H et L sont indépendantes les unes des autres, donc la quantité d'information qu'elles contiennent s'additionne.
   — lat_d et $long_d$ prennent 91 valeurs possibles, équiprobables, donc apportent chacune $\log_2 91 \approx 6{,}508$ bits.
   — lat_m, $long_m$, lat_s et $long_s$ prennent 60 valeurs possibles, équiprobables, donc apportent chacune $\log_2 60 \approx 5{,}907$ bits.
   — H et L prennent deux valeurs possibles, équiprobables, donc apportent chacune 1 bit.

   Au total, on obtient donc 38,643 bits d'information.

   Note : L'énoncé ne précise pas que des positions telles que $0\,^\circ\,0\,'\,0\,''$ N $0\,^\circ\,0\,'\,0\,''$ E et $0\,^\circ\,0\,'\,0\,''$ S $0\,^\circ\,0\,'\,0\,''$ W sont égales, donc on les considère comme étant distinctes.

   (b) Dans ce cas, les variables ne sont pas toutes indépendantes, car lorsque $lat_d = 50$, la valeur de lat_m et de lat_s doit nécessairement être nulle.

— Il y a $20 \times 60 \times 60 + 1 = 72001$ valeurs possibles pour le triplet de variables (lat_d, lat_m, lat_s), toutes équiprobables. Ces variables apportent donc collectivement $\log_2 72001 \approx 16{,}136$ bits.

— La variable $long_d$ apporte toujours 6,508 bits, et les variables $long_m$ et $long_s$ 5,907 bits chacune.

— La variable H possède une valeur fixe, et n'apporte donc pas d'information.

— La variable L apporte toujours un bit.

Au total, on obtient donc 35,457 bits d'information.

2. (a) Le nombre 2021 est tel que $2^{10} < 2021 < 2^{11}$. Pour que -2021 appartienne à l'intervalle $[-2^{n-1} + 1, 2^{n-1} - 1]$ (resp. $[-2^{n-1}, 2^{n-1} - 1]$) des valeurs représentables sur n bits par valeur signée et complément à un (resp. par complément à deux), la plus petite valeur de n pouvant être choisie est donc 12. Cette valeur est identique pour les trois représentations.

(b)

17,625 :		0	0	0	1	0	0	0	1	1	0	1	0
−21,5 :	+	1	1	1	0	1	0	1	0	1	0	0	0
−3,875 :		1	1	1	1	1	1	0	0	0	0	1	0

(c) — Ce nombre est négatif, donc son bit de signe est égal à $\boxed{1}$.

— On a $2^{43} < 3 \cdot 2^{42} < 2^{44}$, donc on choisit un exposant égal à 43 de façon à obtenir une mantisse normalisée. La représentation de cet exposant correspond à l'encodage non signé sur 8 bits de $127 + 43 = 170$, c'est-à-dire $\boxed{10101010}$.

— Pour cette valeur de l'exposant, la mantisse est égale à $\dfrac{3 \cdot 2^{42}}{2^{43}} = 1{,}5$, dont l'encodage est $\boxed{10000000000000000000000}$.

En résumé, on obtient $\boxed{1\,10101010\,10000000000000000000000}$.

(d) Non, car le complément à deux d'un nombre négatif v est égal au complément à un de $v + 1$. Par conséquent, pour toute suite w de bits commençant par un bit de signe égal à 1, on a $[w]_{c_2} = [w]_{c_1} - 1$. Les nombres représentés par w par complément à un et par complément à deux ne peuvent donc pas être égaux.

3. Le compteur de programme est un registre du processeur qui contient en permanence l'adresse dans la mémoire de programme de la prochaine instruction devant être exécutée. Lorsque le processeur exécute une instruction donnée d'un programme, ce registre contient l'adresse de l'instruction suivante, ou bien celle de la destination dans le cas d'une instruction de saut ou d'appel de fonction.

4. (a) Une stratégie possible consiste à mémoriser la valeur du dernier élément du tableau dans une variable `v`, et puis explorer successivement tous les éléments précédents jusqu'à trouver la première valeur différente de `v`, ou atteindre le début du tableau. On obtient le code C suivant.

```
unsigned nb_valeurs(short p[], unsigned n)
{
  unsigned i;
  short v;

  v = p[n - 1];
  i = 1;

  while (i < n && p[n - i - 1] == v)
     i++;

  return i;
}
```

(b) Par souci de facilité, on choisit de représenter `n` et `i` sur 64 bits plutôt que 32. Les variables du programme obtenu en (a) sont associées de la façon suivante aux registres du processeur :

`p` : `RDI`
`n` : `RSI`
`i` : `RAX`
`v` : `CX`

En outre, on utilise le registre `RDX` pour calculer l'index des éléments à lire dans le tableau `p`. On obtient alors le code suivant.

```
            .intel_syntax noprefix
            .text
            .global nb_valeurs
            .type nb_valeurs, @function

nb_valeurs: PUSH  RBP
            MOV   RBP, RSP
            MOV   RDX, RSI
            MOV   CX, word ptr[RDI + 2 * RDX - 2]
            MOV   RAX, 1
boucle:     CMP   RAX, RSI
            JAE   fin
            DEC   RDX
            CMP   CX, word ptr[RDI + 2 * RDX - 2]
            JNE   fin
            INC   RAX
            JMP   boucle
fin:        POP   RBP
            RET
```