Prénoms	Nom	Note

Exercice 1 (4 points)

Pour ce problème, nous utilisons une machine x86-64 bits, little endian. L'état actuel de la mémoire (valeurs en hexadécimal) est présenté ci-dessous :

Word Addr	+0	+1	+2	+3	+4	+5	+6	+7
0x00	AC	AB	03	01	BA	5E	BA	11
0x08	5E	00	68	0C	BE	A7	CE	FA
0x10	1D	B0	99	DE	AD	60	BB	40
0x18	14	1D	EC	AF	EE	FF	CO	70
0x20	BA	B0	41	20	80	AA	BE	EF

char* charP = 0x12

int* intP = 0x8

long* longP = 0x30

En utilisant les valeurs indiquées ci-dessus, complétez le code C ci-dessous pour accomplir les comportements décrits dans les commentaires en utilisant l'arithmétique des pointeurs.

char v1 = *(charP + _____); **// Compléter pour que v1 = 0xAF**

int* v2 = &intP[___]; **// Compléter pour que v2 = 0x14**

Exercice 2 (4 points)

Donner la valeur de la variable i après l'exécution de ce code : (ce qui sera affiché dans l'écran à l'exécution) :

```
#include <stdio.h>
int main(void) {
  for (int i = 3; i <= 6; i+=2)
    switch (--i){
      case 3: case 6: i--;
      case 4: i+=2;
      case 2: i++; break;
      default: i = 0;
    }
  printf(" Ecran : %d ", --i);
  return 0;
}
```

Ecran : _____

Exercice 3 (4 points)

Donner les valeurs affichées pour les deux variables x et y après l'exécution du code ci-dessous :

```
#include <stdio.h>
int main(void) {
  for (int i = 0; i < 3; ++i) {
    int x = 1; static int y = 1;
    ++x;
    y++;
    if(i == 2)  printf("x  = %d - y =  %d\n", x, y);
  }
  return 0;
}
```

x = ____ - y = ____

Exercice 4 (4 points)

Pour ce problème, nous utilisons une machine x86-64 64 bits (little endian). L'état actuel de la mémoire (valeurs en hexadécimal) est affiché ci-dessous (int 4 octets, short 2 octets, long 8 octets) :

Word Addr	+0	+1	+2	+3	+4	+5	+6	+7
0x00	BD	28	ED	02	35	72	3A	AF
0x08	66	6F	B1	E9	00	FF	5D	4D
0x10	86	06	04	30	64	31	8C	B3
0x18	63	78	1E	1C	25	34	EE	93
0x20	42	6C	65	67	DE	AD	BE	EF
0x28	CA	FE	D0	0D	1E	93	FA	CE

Écrivez la valeur en hexadécimal de chaque expression dans les lignes commentées à leur état respectif lors de l'exécution du programme donné. Écrivez INCONNU dans la case vide si la valeur ne peut pas être déterminée.

```
int main(int argc, char** argv) {
    char *charP;
    short *shortP;
    int *intP = 0x00;
    long *longP = 0x28;

    // The value of intP is:              0x_____

    // *intP                              0x_____

    // &intP                              0x_____

    // longP[-2]                          0x_____

    charP = 0x20;
    shortP = (short *) intP;
    intP++;
    longP--;

    // *shortP                            0x_____

    // *intP                              0x_____

    // *((int*) longP)                    0x_____

    // (short*) (((long*) charP) - 2)     0x_____
}
```

Exercice 5 (4 points)

Donner le contenu du tableau **a1** après l'exécution de ce code (remplir la table ci-dessous) :

```
void mystery(int a1[], int length1, int a2[], int length2){
  for (int i = 0; i < length1; i++)
    a1[i] += a2[length2 - i - 1];
}
```

```
int main(void) {
  int a1[] = {1, 3, 5, 7, 9};    int a2[] = {1, 4, 9, 16, 25};
  mystery(a1, 5, a2, 5);         return 0;
}
```

a1[0]	a1[1]	a1[2]	a1[3]	a1[4]