POLYTECH – ELEC3 2022–2023

Contrôle de Info C

Durée : 1h30 Aucun document autorisé

Nom : Prénom :

Notez que les affirmations (antécédents, conséquents, rôles, et invariants) dans vos codes C entreront pour partie dans la note finale.

▶ 1. Expliquez de façon claire comment sont organisées les données d'un programme au cours de son exécution dans la mémoire centrale.

▶ 2. Expliquez de façon claire comment un programme C peut accéder aux *paramètres programmes* fournis au moment du lancement de son exécution.

► 3. Écrivez ce qu'affichent les 10 appels de la procédure `trace` dans la programme suivant :

```c
#include <stdio.h>
#include <stdlib.h>

void trace(const int temp, const int block[]) {
  printf("temp = %d", temp);
  printf(" - bloc = [%d, %d, %d]\n", block[0], block[1], block[2]);
}

int main(void)
{
  int block[3] = {0, 3, 5};
  int *ptr = &block[1];
  int temp;

  temp = block[1];      trace(temp, block);
  temp = *(block + 1); trace(temp, block);
  temp = *(ptr - 1);    trace(temp, block);
  temp = *ptr;          trace(temp, block);

  ptr = block+1;        trace(temp, block);
  temp = *ptr;          trace(temp, block);
  temp = *(ptr+1);      trace(temp, block);

  ptr = block;
  temp = *ptr++;        trace(temp, block);
  temp = ++*ptr;        trace(temp, block);
  temp = *ptr--;        trace(temp, block);

  return EXIT_SUCCESS;
}
```

▶ 4. Écrivez en C la fonction booléenne `estPrefixe` qui teste si une chaîne de caractères `s1` est préfixe d'une chaîne de caractères `s2`. Les deux chaînes sont paramètres de la fonction. Par exemple, *`"abri"`* est préfixe de *`"abricot"`*. Deux chaînes identiques ne sont pas préfixes l'une de l'autre. Vous utiliserez exclusivement la *notation de pointeur*, et aucune fonction de `string.h`.

▶ 5. Soit la déclaration suivante du type `Liste` vu en TD.

```
typedef int T;

typedef struct noeud {
  T elt;
  struct noeud *suivant;
} *Liste;
```

Écrivez à l'aide de la procédure du TD `supprimer`, la procédure `supprimerEnQueue` qui supprime le dernier élément d'une liste. Son en-tête est le suivant :

```
/*
 * Rôle : supprime le dernier élément de la liste
 */
void supprimerEnQueue(Liste *l)
```

▶ 6. Sans utiliser la procédure du TD `supprimer`, récrivez la procédure `supprimerEnQueue` précédente.

▶ 7. À l'aide de la fonction `ieme` (vue en TD), écrivez la procédure `ecrireListe` qui prend en paramètre une liste de type `Liste` et qui écrit tous ses éléments sur la sortie standard. On considérera que la liste contient des entiers (type **int**).

▶ 8. Que pensez-vous de l'efficacité de cette procédure ? Expliquez.