DAEMONLAB

EXAMEN

RED ALERT

Analyse grammaticale

- DaemonLab -

INDEX

01 – Avant-propos

Votre travail doit être rendu via le dossier **~/exam/ra_soviet/** dans votre espace personnel.

SI VOUS FAITES ERREUR ET QUE LE DOSSIER QUE VOUS UTILISEZ POUR VOTRE RENDU EST DIFFERENT, VOUS NE SEREZ PAS EVALUE FAUTE D'AVOIR PU TROUVER VOTRE TRAVAIL.

Ce travail est a effectuer seul. De plus il s'agit d'un examen. Vous n'avez donc pas le droit de communiquer avec vos camarades.

Votre rendu doit respecter **strictement** l'ensemble des règles suivantes :

- Il ne doit contenir **aucun** fichier objet. (*.o)
- Il ne doit contenir **aucun** fichier tampon. (*~, #*#)
- Il ne doit pas contenir votre production finale (programme ou bibliothèque)

LA PRESENCE D'UN FICHIER INTERDIT METTRA IMMEDIATEMENT FIN A VOTRE EVALUATION !

Votre programme doit respecter les Tables de la Norme dans leur intégralité. Vous êtes invité à les observer depuis **l'Infosphère**. Elles sont disponibles comme ressource de cette activité.

02 – Fonctions autorisées

La bibliothèque logicielle venant avec le C est vaste et disponible. La LibLapin, que vous utilisez dans vos projets multimédia, est également vaste… Cependant nous avons fait le choix de vous interdire leurs utilisation intégrales, afin de vous amener progressivement à reprogrammer vous même ses fonctionnalités les plus utiles.

L'UTILISATION D'UNE FONCTION INTERDITE EST ASSIMILEE A DE LA TRICHE. LA TRICHE PROVOQUE L'ARRET DE L'EVALUATION ET LA PERTE DES MEDAILLES.

Vous n'avez le droit d'utiliser aucune fonction issue de la LibC ou de la LibLapin à l'exception de celles que nous vous autoriserons explicitement.

POUVOIR UTILISER UNE FONCTION NE SIGNIFIE PAS NECESSAIREMENT QUE CELLE-CI SOIT UTILE A VOTRE CAS.

Pour cette activité, issu de la LibC, vous n'avez le droit qu'à la liste suivante :

- open
- close
- read
- srand
- cos
- atan2
- time
- write
- alloca
- atexit
- rand
- sin
- sqrt
- malloc, free

03 – Parcours d'espace

Fichier read_space.c

« VOUS ALLEZ DEVOIR FAIRE FACE A UNE FORTE ADVERSITE, MAIS RASSUREZ VOUS, VOUS AVEZ LE SOUTIEN DU NKVD ! UN POSTE DE TRANSMISSION A ETE PLACE PAR UN CAMARADE SUR LA RIVE OUEST. LES ALLIES NE DEVRAIENT PAS POUVOIR VOUS ARRETER SI VOUS RESTEZ SUR LE CHEMIN QUI A ETE PREVU POUR VOUS. TERMINE. »

Implémentez la fonction suivante :

```
int        std_read_space(const char        *str,
                          size_t            index)
```

Cette fonction lit dans **str** à partir de **str[index]** les caractères espace, saut de ligne, retour chariot et tabulation et les dépasse. Elle renvoi le nombre de caractères parcourus.

04 – Lecture d'entier

Fichier read_base10.c

« L'AVANCEMENT EST RAPIDE DANS L'ARMEE ROUGE. CONTINUEZ A ACCUMULER LES VICTOIRES ET LE FUTUR SERA A VOUS ! VOTRE MISSION MAINTENANT QUE VOUS AVEZ ACQUIS LE POSTE DE TRANSMISSION SERA DE DECODER UNE PREMIERE TRANSMISSION PROVENANT DU POLITBURO. IL S'AGIRA D'UNE SUITE DE CHIFFRE DEVANT ETRE INTERPRETE COMME ETANT UN SEUL ET UNIQUE NOMBRE. TERMINE.»

Implémentez la fonction std_read_base10 :

```
int             std_read_base10(const char    *str,
                                size_t        index,
                                int           *out);
```

La fonction read_base10 lit dans str à partir de str[index] des chiffres et renvoi le nombre qu'ils forment. La fonction read_base10 s'arrête dès qu'un caractère n'étant pas un chiffre est lu.

Le nombre de caractères lu est renvoyé. Le nombre lu par la fonction est stocké dans *out. Si out vaut NULL, la fonction n'y stockera rien.

Ci-dessous, un main de test. Ne le rendez pas.

```
int       main(void)
{
     int  out;
     int  rd;

     rd = std_read_base10("abc123", 3, &out);
     if (rd != 3 || out != 123)
          tc_putchar('n'); // NON!
     else
          tc_putchar('o'); // Oui! <3
     return (0);
}
```

05 – Lecture de tableau

Fichier read_array_base10.c

« FELICITATION, CAMARADE : L'UNION DES REPUBLIQUES SOCIALITES SOVIETIQUES EN ENTIER EST FIER DE VOUS. CE PREMIER NOMBRE N'ETAIT QU'UN TEST AFIN DE VERIFIER LA VIABILITE DU SYSTEME ET DE VOTRE DECODAGE. LA VERITABLE TRANSMISSION APPROCHE. UNE SERIE DE NOMBRE CHACUN SEPARE PAR UNE UNIQUE VIRGULE. J'ENTENDS LES SIRENES S'ACTIVER. LES ALLIES SAVENT QUE NOUS SOMMES ICI. ILS RISQUENT DE BROUILLER LA TRANSMISSION, FAITES VITE !TERMINE. »

Implémentez la fonction suivante :

```
int       std_read_array_base10(const char   *str,
                                size_t       index,
                                int          *out,
                                int          *nbr_out)
```

Cette fonction lit dans **str** à partir de **str[index]** des séries de chiffres séparés par une virgule. Le paramètre **out** est un tableau servant à stocker ces valeurs. Le paramètre **nbr_out** est un espace permettant de stocker le nombre d'entier stocker dans **out**.

Si **out** ou **nbr_out** vaut **NULL**, les nombres ne seront pas stockés. Si out seul est **NULL**, le nombre de chiffre lisible est rangé dans **nbr_out**. Il sera toujours supposé que **out** est suffisament grand pour contenir tous les entiers.

Ci-dessous, un main de test. **Ne le rendez pas.**

```
int       main(void)
{
     int  nbr_out;
     int  out[10];
     int  rd;

     rd = std_read_array_base10
        ("abc123,456,789", 3, &out[0], &nbr_out);
     if (rd != 11 || nbr_out!= 3 || out[0] != 123
          || out[1]!= 456 || out[2]!= 789)
          tc_putchar('n'); // NON!
     else
          tc_putchar('o'); // Oui! <3
     return (0);
}
```

06 – Split

Fichier split.c

« Camarade ! Les futures informations vont arriver sous des formes plus variees. Le Soviet Supreme vient de decreter l'ouverture de l'exploitation de l'entier R.S.K.O.I* ! Vous allez donc devoir permettre de stocker n'importe quelle information, separee les uns des autres par un separateur parametrable ! »

Implémentez la fonction suivante :

```
char            **std_split(const char   *s,
                            char         separator)
```

Cette fonction alloue un tableau de chaîne de caractères contenant les caractères situés entre les instances du séparateur passé en paramètre. Par exemple, "abc,def,ghi" avec virgule comme séparateur va renvoyer un tableau de 4 cases contenant les trois chaînes suivi d'un pointeur NULL.

```
int       main(void)
{
     char **wt;

     wt = std_split("abc,def,g,", ',');
     if (strcmp(wt[0], "abc") != 0 ||
         strcmp(wt[1], "def") != 0 ||
         strcmp(wt[2], "g") != 0 ||
         strcmp(wt[3], "") != 0 ||
         wt[4] != NULL)
          tc_putchar('n'); // NON!
     else
          tc_putchar('o'); // Oui! <3
     return (0);
}
```

** RSKOI est l'acronyme de « Russian standard code for information interchange » traduit en russe. Ceci est donc une référence de qualité à l'ASCII, American standard code for information interchange. Riez intérieurement, s'il vous plaît.*