Drugi kolokvijum iz Programiranja 2

Kolokvijum traje 90 minuta

Napomene:
a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
b) Vrednost odgovora: tačan = **15**; netačan = **-3.75**; nevažeći (nula ili više zacrnjenih kružića) = **0.**
c) Na pitanjima se može osvojiti najviše **45** poena.
d) Zadatak nosi **55** poena.

I ZADACI

1)Napisati program na programskom jeziku C koji vrši određenu obradu nad stringovima. Program najpre treba da pročita matricu karaktera dimenzija N x 2 koja predstavlja preslikavanje karaktera u cifru. Program prvo treba da pročita broj redova date matrice, a zatim i same elemente matrice. Svaka vrsta matrice predstavlja jedno preslikavanje i to tako da prva kolona predstavlja sam karakter, a druga kolona cifru u koju se preslikava dati karakter. Potom program treba da pročita tri reči, da izvrši preslikavanje reči u brojeve na osnovu matrice preslikavanja tako što svaki karakter menja odgovarajućom cifrom i proveri da li važi **`PRVA_REČ+DRUGA_REČ=TREĆA_REČ`**. Ukoliko ova jednakost važi potrebno je na standardnom izlazu ispisati „**`DA`**“, a u suprotnom „**`NE`**“. Program realizovati prema sledećim stavkama:

- Napisati funkciju **`char** loadMap(int rows)`** koja formira matricu karaktera dimenzija **`rows x 2`** i popunjava je po vrstama karakterima koje čita iz jednog reda sa standardnog ulaza.
- Napisati funkciju **`int charToDigit(char c, char **map, int rows)`** koja karakter **`c`** preslikava u odgovarajuću cifru na osnovu matrice preslikavanja **`map`**. Dimenzije matrice preslikavanja su **`rows x 2`**. Ukoliko ne postoji preslikavanje za odgovarajući karakter vratiti 0.
- Korišćenjem prethodno realizovane funkcije realizovati funkciju **`int stringToInt(char *word, char **map, int rows)`** koja string **`word`** preslikava u ceo broj tako što svaki karakter iz stringa **`word`** preslika u odgovarajuću cifru na osnovo matrice preslikavanja **`map`**.
- Napisati funkciju **`char* readLine()`** koja čita jedan red sa standradnog ulaza. Prilikom čitanja zauzeti samo onoliko prostora koliko je neophodno za smeštanje svih karaktera jednog reda.
- Korišćenjem prethodno realizovanih funkcija napisati glavni program koji prvo čita broj vrsta matrice preslikavanja, potom i same elemente matrice, a nakon toga tri reči, svaku u zasebnom redu, i na kraju ispisuje „**`DA`**“ ukoliko data jednakost važi, a „**`NE`**“ u suprotnom slučaju.

Ulaz:	Izlaz:	Objašnjenje
8 O 0 M 1 Y 2 E 5 N 6 D 7 R 8 S 9 SEND MORE MONEY	DA	SEND = 9567 MORE = 1085 MONEY = 10652

II PITANJA

1)Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
#include <string.h>
#define MAX 15
void f1(char s[][MAX], char **p){
  char *d = &s[0][0];
  for (int i = 0; ; i++) {
    p[i] = d;
    if (!strlen(d)) break;
    d += MAX;
  }
}
void f2(char **p) {
 char str[MAX];
 int n = 2, c;
 do {
  c = 0; int d = -1;
  for (int i=0;strlen(p[i]);i++){
    if (strlen(p[i]) >= n) {
      c++;
      if (d < 0) memcpy(str, p[i], n);
      else memcpy(p[d], p[i], n);
      d = i;
    }
  }
  if (d >= 0) memcpy(p[d], str, n);
  for (int i = 0; strlen(p[i]); i++) {
    if(strlen(p[i]) >= n) p[i] += n;
  }
 } while (c);
}
int main() {
 char str[][MAX]={"April","2018/19","Kolokvijum", ""};
  char *p[sizeof(str) / MAX];
  f1(str, p);
  f2(p);
  printf("%s\n", &str[2][0]); return 0;
}
```

(A) Apri/1ijum B) Ap20kvijum C) Aprikvijum

2)Šta ispisuje program na programskom jeziku C ukoliko funkcije **writeMatrix** i **freeMatrix** rade redom ispis matrice po vrstama i oslobađanje dinamičke memorije koju je zadata matrica zauzimala?

```
#include <stdio.h>
#include <stdlib.h>
void freeMatrix(int** a, int n, int m);
void writeMatrix(int** a, int n, int m);
int** createMatrix(int n, int m){
  int** a, k = 1;
  a = malloc(n * sizeof(int*));
  for(int i = 0; i < n; i++){
    a[i] = malloc(m * sizeof(int));
    for(int j = 0; j < m; j++) a[i][j] = k;
        k++;
    } return a;
}
```

```
void f(int **a, int count) {
    int* t = a[count-1];
    for(int i = count-1; i > 0; i--) {
        a[i] = a[i-1];
    } a[0] = t;
}
int main(void) {
  int n = 3, m = 4;
  int** a = createMatrix(n, m); f(a, n);
  writeMatrix(a,n,m); freeMatrix(a,n,m);
}
```

(A) 3 3 3 3
1 1 1 1
2 2 2 2

B) 2 2 2 2
3 3 3 3
1 1 1 1

C) 3 3 3 3
1 1 1 1
1 1 1 1

3)Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
#include <stdlib.h>
#define N 4
#define M 2
void main() {
  int i, j, **b, s=0;
  b = malloc(N * sizeof(int***));
  for (i = 0; i < N; i++) {
  b[i] = calloc(i<M?i+1:i-M+1, sizeof(int));
  for (j = 0; j < (i<M?i+1:i-M+1); j++)
    (*(b+i))[j] = s++;
  }
for (i = N - 1; i >= 0; i--) {
  for (j = 0; j < (i<M?i+1:i-M+1); j++)
    printf("%d ", *(b[i]+j));
  free(b[i]);
  }
  free(b);
}
```

A) 6 5 4 3 2 1 0

(B) 4 5 3 1 2 0

C) 4 5 6 2 3 1 0

4)Koja od sledećih tvrđenja su tačna za sledeći program na programskom jeziku C koji ispisuje dužinu unetog stringa?

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define MAX 10
int main() {
  char *s1, *s2, i = 0, c;
  s1=(char*)malloc((MAX + 1)* sizeof(char));
  if (!s1) return 1;
  while (((c = getchar())!='\n') && i < MAX)
    s1[i++] = c;
  s1[i] = '\0';
  s2 =
realloc(s1,(strlen(s1)+1)*sizeof(char));
  if (!s2) {
    free(s1);
    return 1;
  } else s1 = s2;
  printf("%d\n", strlen(s1));
  free(s1); return 0;
}
```

(A) Naredba `s1=(char*)malloc((MAX + 1)*sizeof(char));` se može zameniti sa `s1 = realloc(0L, MAX + 1);`

B) Naredba `s2 = realloc(s1, (strlen(s1) + 1) * sizeof(char));` se može zameniti sa `free(s1 + (strlen(s1) + 1);`

C) Program se neće završiti korektno ukoliko korisnik unese više od MAX karaktera.

SI1P2, Programiranje 2, Drugi kolokvijum, 2019.
Rešenje zadatka

Zadatak

Napisati program na programskom jeziku C koji vrši određenu obradu nad stringovima. Program najpre treba da pročita matricu karaktera dimenzija N x 2 koja predstavlja preslikavanje karaktera u cifru. Program prvo treba da pročita broj redova date matrice, a zatim i same elemente matrice. Svaka vrsta matrice predstavlja jedno preslikavanje i to tako da prva kolona predstavlja sam karakter, a druga kolona cifru u koju se preslikava dati karakter. Potom program treba da pročita tri reči, da izvrši preslikavanje reči u brojeve na osnovu matrice preslikavanja tako što svaki karakter menja odgovarajućom cifrom i proveri da li važi **`PRVA_REČ+DRUGA_REČ=TREĆA_REČ`**. Ukoliko ova jednakost važi potrebno je na standardnom izlazu ispisati „**`DA`**“, a u suprotnom „**`NE`**“. Program realizovati prema sledećim stavkama:

- Napisati funkciju **`char** loadMap(int rows)`** koja formira matricu karaktera dimenzija **`rows x 2`** i popunjava je po vrstama karakterima koje čita iz jednog reda sa standardnog ulaza.
- Napisati funkciju **`int charToDigit(char c, char **map, int rows)`** koja karakter **`c`** preslikava u odgovarajuću cifru na osnovu matrice preslikavanja **`map`**. Dimenzije matrice preslikavanja su **`rows x 2`**. Ukoliko ne postoji preslikavanje za odgovarajući karakter vratiti 0.
- Korišćenjem prethodno realizovane funkcije realizovati funkciju **`int stringToInt(char *word, char **map, int rows)`** koja string **`word`** preslikava u ceo broj tako što svaki karakter iz stringa **`word`** preslika u odgovarajuću cifru na osnovo matrice preslikavanja **`map`**.
- Napisati funkciju **`char* readLine()`** koja čita jedan red sa standradnog ulaza. Prilikom čitanja zauzeti samo onoliko prostora koliko je neophodno za smeštanje svih karaktera jednog reda.
- Korišćenjem prethodno realizovanih funkcija napisati glavni program koji prvo čita broj vrsta matrice preslikavanja, potom i same elemente matrice, a nakon toga tri reči, svaku u zasebnom redu, i na kraju ispisuje „**`DA`**“ ukoliko data jednakost važi, a „**`NE`**“ u suprotnom slučaju.

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

#define NUMBER_OF_COLUMNS 2
char** loadMap ( int *rows ) {
    char **map = NULL;
    scanf ( "%d", rows );

    map = ( char** ) calloc ( *rows, sizeof ( char* ) );
    if ( map == NULL ) {
        exit ( 1 );
    }

    for ( int i = 0; i < *rows; ++i ) {
        map[i] = ( char* ) calloc ( NUMBER_OF_COLUMNS, sizeof ( char ) );
        if ( map[i] == NULL ) {
            exit ( 1 );
        }

        scanf ( " %c %c", &map[i][0], &map[i][1] );
    }

    getchar ( );

    return map;
}
int charToDigit ( char c, char **map, int rows ) {
    for ( int i  = 0; i < rows; ++i ) {
        if ( map[i][0] == c ) {
            return map[i][1] - '0';
        }
    }

    return 0;
}
int stringToInt ( char *string, char **map, int rows ) {
    int number = 0;

    for ( int i = 0; i < strlen ( string ); ++i ) {
        number = number * 10 + charToDigit ( string[i], map, rows );
    }

    return number;
}
char *readLine ( ) {
    char *line = NULL;
    int size   = 0;

    while ( 1 ) {
        char c = getchar ( );

        char *temp = realloc ( line, ( size + 1 ) * sizeof ( char ) );
        if ( temp == NULL ) {
            exit ( 1 );
        }
        line = temp;

        if ( c != '\n') {
            line[ size ] = c;
            size++;
        } else {
            line[size] = '\0';
            break;
        }
    }

    return line;
}
```

```c
int main ( ) {
    int rows;
    char **map        = loadMap ( &rows );
    char *firstWord  = readLine ( );
    char *secondWord = readLine ( );
    char *thirdWord  = readLine ( );

    int firstNumber  = stringToInt ( firstWord, map, rows );
    int secondNumber = stringToInt ( secondWord, map, rows );
    int thirdNumber  = stringToInt ( thirdWord, map, rows );

    if ( ( firstNumber + secondNumber ) == thirdNumber ) {
        printf ( "DA" );
    } else {
        printf ( "NE" );
    }

    for ( int i = 0; i < rows; ++i ) {
        free ( map[i] );
    }
    free ( map );
    free ( thirdWord );
    free ( secondWord );
    free ( firstWord );

    return 0;
}
```