ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU BEOGRAD, 09.07.2014.

Ispit iz Programiranja 2

Ispit traje 150 minuta

Napomene:
a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
b) Vrednost odgovora: tačan = **5**; netačan = **-1.25**; nevažeći (nula ili više zacrnjenih kružića) = **0.**
c) Na pitanjima se može osvojiti najviše **25** poena.
d) Svaki zadatak nosi **20** poena.

I ZADACI

1)Napisati program na programskom jeziku C za izračunavanje plata radnika u skladu sa njihovim učinkom. Učinak svakog radnika se zadaje kao jedan red matrice. Element matrice (*i,j*) je vreme zapisano kao realan broj sati koje je radnik *i* proveo radeći na aktivnosti *j*. Cene radnih sati za aktivnosti se zadaju nizom realnih brojeva. Element sa indeksom *j* je cena radnog sata aktivnosti *j*. Na početku, program sa standardnog ulaza pročita broj radnika, kao i broj različitih aktivnosti, a potom učita niz sa cenama radnih sati aktivnosti i matricu sa učinkom radnika. Potom, program izračuna plate svih radnika i iz niza izbaci plate koje su ispod prosečne plate za taj mesec. Na kraju, program ispiše iznose plata koje nisu izbačene iz niza. Za potrebe računanja plata napisati potprogram koji kao rezultat vraća niz sa iznosima plata svih radnika. Program sa potprogramom komunicira isključivo preko argumenata i povratne vrednosti potprograma. Prilikom implementacije nije dozvoljeno koristiti pomoćne nizove. Voditi računa o pravilnom rukovanju korišćenim resursima.

2)Napisati program na programskom jeziku C za poređenje dva teksta. U datotekama *original.txt* i *kopija.txt* se nalaze tekstovi koji se porede. Program na standardnom izlazu treba da ispiše broj reči koje su iste u oba teksta, broj reči koje postoje u originalu ali ne postoje u kopiji i broj reči koje postoje u kopiji ali ne postoje u originalu. Ukoliko se jedna reč pojavi N puta u originalu i M puta u kopiji, računa se na sledeći način. Ukoliko je N jednako M, tada se računa da je to N istih reči. Ukoliko je N veće od M, tada se računa da ima M istih reči, dok original ima N-M reči koje ne postoje u kopiji. Analogno, ako je N manje od M, tada ima N istih reči, dok ima M-N reči koje se pojavljuju samo u kopiji. Reč je neprekidni niz od maksimalno 30 slova. Prilikom poređenja reči zanemariti razliku u malim i velikim slovima. Voditi računa o pravilnom rukovanju korišćenim resursima. Nije dozvoljeno ponavljati čitanje bilo koje od dve datoteke. Primer ulaza i izlaza:

original.txt	kopija.txt	stdout
Danas nam je divan dan, Aleksandri je rodjendan, zivela i srecna nam bila.	Danas nam je divan dan, divan dan. Srecan rodjendan. Zivela!	isto - 7 original - 6 kopija - 3

II PITANJA

1)Šta je rezultat izvršavanja sledećeg potprograma ako se pozove sa `f(matrica,k,k)`? Pretpostaviti da je `matrica` ispravno alocirana i inicijalizovana dinamička kvadratna matrica dimenzija `n x n` i da je `k<n`.

```
int f(int** mat, int i, int j) {
  if (i>0)
    if (j>i)
      return f(mat, i, j-1);
    else return mat[j][i]+f(mat,i-1,j);
  else return mat[i][i]; }
```

A) Sabira elemente glavne dijagonale počev od elementa (k,k), zaključno sa elementom (n-1,n-1).
(B) Sabira elemente glavne dijagonale počev od elementa (0,0), zaključno sa elementom (k,k).
C) Sabira elemente glavne dijagonale počev od elementa (0,0), zaključno sa elementom (k,k), kao i elemente koji su neposredno ispred odgovarajućih elemenata glavne dijagonale.

2)Koja tvrdnja je uvek tačna za listu koja se dobije kao rezultat izvršavanja potprograma `obradi` kojem se prosledi neprazna lista proizvoljnog sadržaja?

```
typedef struct elem {
  int broj;
  struct elem *sled;
} Elem;
Elem* obradi(Elem* lista){
  Elem* pre = lista, tek = lista->sled;
  while (tek != NULL) {
    if (tek->broj < pre->broj) {
      pre->sled = tek->sled;
      free(tek);
    } else pre = tek;
    tek = pre->sled; } }
```

A) Elementi liste su uređeni po strogo rastućem poretku.
B) Lista ne sadrži iste vrednosti u susednim elementima.
(C) Elementi liste su uređemo po neopadajućem poretku.

3)Koje od ponuđenih tvrdnji su tačne za programski jezik C?
A) `str1 = str2;` je potpuno ispravna dodela vrednosti na programskom jeziku C ako su `str1` i `str2` deklarisani kao `char str1[10], str2[10].`
(B) `char *((* (* x) (int, double))[10][10]);` je potpuno ispravna deklaracija na programskom jeziku C.
C) Ako je deklarisana struktura `struct str {int polje;};` i pokazivač `struct str* pstr`, tada je pristup polju strukture moguć preko `*pstr.polje`.

4)Šta ispisuje sledeći program napisan na programskom jeziku C?

```
#include <stdio.h>
int main(){
    char single[] = "ispitivaci", bunch[] = "JULSKI ispitni rok";
    char *n = single, *h = bunch;
    for(; *n && *h; h++)
        if(*h==*n) n++;
        else if(n-single) break;
    printf("%ld\n", n-single); }
```

(A) 5 B) 1 C) 10

5)Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
#define X(a) sizeof(a)/sizeof(int)
typedef int (*FP)(int*);
int f1(int*); int f2(int*); int f3(int*);
FP f[] = {f1, f3, f2, f3, f3, f1};
int f1(int *p) {
   FP t = f[*p%5]; f[*p%5] = f[(*p+3)%5];
   f[*p = (*p+3)%5] = t; return *(p -= 4); }
int f2(int *q) {
   return *(q-3) = (*f[(*q)%5])(q-2); }
int f3(int *r) { return ++r[-2]; }
void main() {
   int a[] = {4, 8, 5, 3, 7, 4}, n = X(a);
   int i = n, s = 0;
   while (--i > 3) a[i] = f[i](a+i);
   for (i=0; i < n; s += a[i++]);
   printf("%d", s); }
```

A) 35 **(B)** 31 C) 34

6)Šta ispisuje sledeći program na programskom jeziku C, ako je sadržaj argumenata komandne linije (bez naziva programa): `leto je najtoplije`?

```
#include <stdio.h>
#include <string.h>
void main(int argc, char **argv) {
  char str[100], z;
  int i, j, n;
  *str = '\0';
  for (i = 1; i < argc; i++) {
    strcat(str, argv[i] + i);
    n = strlen(str);
    z = str[0];
    for (j = 0; j < n - 1; j++)
      str[j] = str[(j + 1) % n] ;
    str[n - 1] = z;
  }
  printf("%s", str); }
```

A) `oetoplijet` **(B)** `ettoplijeo` C) `oleejetopl`