

Prvi deo ispita iz Programiranja 2Ispit traje **80** minuta**Napomene:**

- a)** Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
b) Vrednost odgovora: tačan = **20**; netačan = **-5**; nevažeći (nula ili više zacrtnjenih kružića) = **0**.
c) Na pitanjima se može osvojiti najviše **60** poena.
d) Zadatak nosi **40** poena.

I. ZADACI

1) Napisati program na programskom jeziku C koji određuje da li je neki od bita od interesa aktivan (ima vrednost 1) u pozitivnim celim brojevima koji se unose sa standardnog ulaza. Smatrati da se celi brojevi predstavljaju na dužini od 16 bita. Program najpre treba da ponudi korisniku da unese niz od tačno 16 znakova 0 ili 1. Uneti niz predstavlja kombinaciju bita od interesa. Program zatim čita ceo broj sa standardnog ulaza i za njega proverava da li je najmanje jedan bit od interesa aktivan. U slučaju da je uslov ispunjen, program ispisuje poruku "USLOV ISPUNJEN" odnosno "USLOV NIJE ISPUNJEN" u suprotnom. Postupak čitanja celog broja i ispitivanja se ponavlja sve dok korisnik ne unese vrednost 0.

Primer: za uneti niz 0110111101110011 i za unet broj 5, program treba da ispiše "USLOV ISPUNJEN" (binarna reprezentacija broja 5 je 000000000000101), a za unet broj 12 program treba da ispiše "USLOV NIJE ISPUNJEN" (binarna reprezentacija broja 12 je 0000000000001100).

II. PITANJA

1) Realni brojevi se smeštaju u n-bitne lokacije, u duhu IEEE standarda za predstavljanje realnih brojeva, a prema sledećem formatu: jedan bit za predznak broja, k bita za eksponent i p bita za predstavljanje normalizovane mantise sa skrivenim bitom ($1 \leq M < 2$). Poznato je da se brojevi k i p biraju tako da među njima važi relacija: $k = \left\lfloor \frac{p}{2} \right\rfloor$. Takođe, poznato je da je za predstavljanje realnih

brojeva upotrebljen najmanji moguć broj bita tako da najveći ceo broj može da se predstavi pomoću realnih brojeva bez gubitka preciznosti. Neka je R realan, a C ceo broj. Ako se celi brojevi predstavljaju na dužini od 12 bita i broj C ima vrednost maxINT, kako izgleda sadržaj lokacije R nakon dodele $R := C$?

(A) $67FF_{16}$ **(B)** $76EF_{16}$ **(C)** $FE80_{16}$

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
main() {
int i, a=3;
for(i=0, a=5; ; i++) {
switch(i) {
case 0: a *= i; break;
case 1: a <= i;
case 2: i++; a = a+i; break;
case 4: ++a; continue;
default: a += a*i;
}
printf("%d", a);
if( i > 3 ) break;
}
printf("%d", i);
}
```

- (A)** 024510 **(B)** 02235 **(C)** 02455

3) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
main() {
int i = 0, j = 0x2A7F;
while( j & 0x5A50 )
++i, j = j & 0x41C ? j >> 1 : j - i;
printf("0x%x", j);
}
```

- (A)** 0x400 **(B)** 0xa9 **(C)** 0xAF

4) Koja od sledećih tvrdjenja su tačna na programskom jeziku C?

- A)** Celobrojne konstante se mogu zadavati samo u decimalnom i heksadecimalnom brojnom sistemu.
B) Uloga direktive #include je da obezbedi detaljne izveštaje o greškama prilikom prevođenja.
(C) Rezultat operatora deljenja / je ceo broj samo ako su oba operanda celobrojni tipovi.

Drugi deo ispita iz Programiranja 2Ispit traje **80** minuta**Napomene:**

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
 b) Vrednost odgovora: tačan = **20**; netačan = **-5**; nevažeći (nula ili više zacrtnjenih kružića) = **0**.
 c) Na pitanjima se može osvojiti najviše **60** poena.
 d) Zadatak nosi **40** poena.

I. ZADACI

1) Napisati program na programskom jeziku C koji pronalazi i izbacuje prvi najduži podniz cifara u unetom znakovnom nizu (stringu). Napisati potprogram koji vrši unos stringa sa glavnog ulaza. Kako dužina stringa nije unapred poznata, memoriju je potrebno alocirati dinamički. Nakon izbacivanja pronađenog podniza iz stringa, izvršiti realokaciju memorije, ispisati rezultujući string i osloboditi zauzetu memoriju. Postupak ponavljati sve dok se ne unese prazan string.

II. PITANJA

- 1)** Koja od sledećih tvrdjenja su tačna na programskom jeziku C?
 A) Dozvoljena je primena pobitskih operatora nad pokazivačkim promenljivama.
 (B) Deklaracija `void **(*(**ε)[])()` je sintaksno ispravna.
 C) Funkcija `malloc` inicijalizuje alociranu dinamičku memoriju na 0.

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
main() {
    char a[4] = { {'M', 'A', 'J', '2'}, {'P', '2'}, {'A', 'V', 'G'} },
    *b = *(a + 1), *c = *(a + 2);
    do { putchar(b[c++ - b]); } while (*b++);
    printf("%s%s\n", a, b, c);
}
```

- (A) AVGMAJ B)AVG C)AVGMAJP2AVG

3) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
main(){
    int *A, B[6], i;
    A = &B[2];
    for (i=0; i<4; i++) A[i] = 1;
    for (i=5; i>=0; i--) B[5-i] = i % 3;
    for (i=0; i<6; i++) A[B[i]] = i;
    for (i=0; i<6; i++) printf("%d ", B[i]);
}
```

- A)0 2 1 5 3 0 B)2 1 5 2 1 1 (C) 2 1 5 3 0 0

4) Funkcija `xx` na jeziku C ovde prikazana:

```
char *xx(char *s) {
    int i = 0;
    char *x = s;
    if( ! s ) return NULL;
    while(s[i] == ' ') i++;
    if( i == 0 ) return x;
    while( *x++ = s[i++] );
    return s;
}
```

- (A) Izbacuje vodeće znake razmaka iz znakovnog niza s i vraća adresu znakovnog niza kao rezultat.
 B) Pronalazi prvi znak u znakovnom nizu s koji se razlikuje od znaka razmaka i vraća njegovu adresu kao rezultat.
 C) Određuje broj pojavljivanja znaka razmak u znakovnom nizu s i vraća adresu poslednjeg pronađenog znaka razmaka.

Treći deo ispita iz Programiranja 2Ispit traje **80** minuta**Napomene:**

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
 b) Vrednost odgovora: tačan = **20**; netačan = **-5**; nevažeći (nula ili više zacrnjenih kružića) = **0**.
 c) Na pitanjima se može osvojiti najviše **60** poena.
 d) Zadatak nose po **40** poena.

I. ZADACI

1) U tekst datoteci "dnevnik.txt" se nalaze podaci o ocenama učenika iz pojedinih predmeta u srednjoj školi. U svakom redu datoteke se nalazi redni broj učenika u dnevniku (ceo broj praćen tačkom), ime i prezime učenika (svako polje do 30 karaktera), naziv predmeta (polje do 20 karaktera) i u ostatku reda celi brojevi od 1 do 5 koji predstavljaju ocene iz datog predmeta, a kojih može biti proizvoljno mnogo. Napisati program na programskom jeziku C koji na osnovu sadržaja datoteke "dnevnik.txt" formira datoteku čije ime se zadaje preko argumenta komandne linije. Nova datoteka treba da se razlikuje od datoteke "dnevnik.txt" jedino po tome što umesto niza ocena treba da stoji zaključna ocena iz predmeta za svakog učenika. Zaključivanje ocena vršiti zaokruživanjem aritmetičke sredine postignutih ocena, osim ukoliko učenik ima neku slabu ocenu (ocena 1), kada mu se zaključuje slaba ocena. Čitanje jednog reda ulazne datoteke realizovati kao poseban potprogram, uz korišćenje odgovarajuće strukture za smeštanje relevantnih podataka. U slučaju neuspešnog otvaranja datoteke, ispisati poruku o grešci i prekinuti izvršenje programa.

II. PITANJA

1) Šta ispisuje sledeći program, ukoliko mu se preko argumenata komandne linije prenese 3 6 8 5 8?

```
#include <stdio.h>
#include <stdlib.h>
struct A { int a,b;};
struct B { int a,b,c;};
union C { struct A c; struct B d; };
int main(int argc, char* argv[]){
    union C e,f; struct B g;
    f.c.a = atoi(argv[1]); f.c.b = atoi(argv[2]);
    g.a = atoi(argv[3]); g.b = atoi(argv[4]); g.c = atoi(argv[5]);
    f.d = g;
    for(e.d.a=0;e.d.a<f.c.b;e.d.a++)
        for(e.c.a=0;e.c.a<5;e.c.a++)
            for(e.c.b=0;e.c.b<5;e.c.b++) f.c.a++;
    printf("%d %d", f.c.a, f.c.b);
    return 0;
}
```

(A)33 5

B) 28 6

C)28 5

2) Šta treba da stoji umesto #### i &&&& da bi sledeća funkcija ispravno dodavala element na kraj jednostruko ulančane liste, uz pretpostavku da alokacije memorije uvek uspeva?

```
#include <stdlib.h>
typedef struct elem { int broj; struct elem *sled; } Elem;

Elem *na_kraj (Elem *lst, int b) {
    Elem *novi = malloc (sizeof(Elem));
    novi->broj = b; novi->sled = NULL;
    if (!lst) return novi;
    else {
        Elem *tek = lst;
        while (####) tek = tek->sled;
        &&&&
        return lst;
    }
}
```

- A) ####: tek B) ####: tek->sled **(C)####: tek->sled**
 &&&&: tek->sled = novi; &&&&: novi->sled = tek; &&&&: tek->sled = novi;

3) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Element {int x; struct Element *next;} ElListe;
void obradi(ElListe *p) {
    ElListe *p1,*p2;
    for (p1=NULL; p1=NULL; p2=p->next,p->next=p1,p1=p,p=p2);
}
main() {
    ElListe *pok=NULL,*pt;
    int i;
    for (i=0; i<5; i++) {
        pt=(ElListe*)malloc(sizeof(ElListe)); pt->x=i; pt->next=pok; pok=pt;
    }
    obradi(pok);
    for (pt=pok; pt!=NULL; pt=pt->next) printf("%d", pt->x);
}
```

A) 01234

B)43210

(C)4

4) Šta ispisuje sledeća funkcija na programskom jeziku C?

```
#include <stdio.h>

void f2(unsigned id) {
    if (id>1)
        f2(id>>1);
    printf("%d",id&1);
}
```

A) koren broja id

(B)binarni oblik broja id

C) binarni oblik kvadrata broja id