

Prvi 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 računa pobednika u nekoj kartaškoj igri. Igra se igra sa 52 karte. Karta je određena vrednošću i znakom. Vrednost može biti ceo broj od 2 do 14. Postoje četiri znaka čije su jednoslovne oznake K, H, T i P. Korisnik u prvom redu na standardnom ulazu unosi broj igrača N koji igraju igru, kao i glavni znak u partiji. Maksimalan broj igrača je 10. Nakon toga se u N redova, za svakog igrača u posebnom redu, unose karte koje poseduje. Broj karata koje jedan igrač ima je 5. Za svaku kartu se unosi vrednost karte i znak, odvojeni tačno jednim razmakom. Karte u redu su međusobno odvojene jednim razmakom. Broj poena koje jedan igrač ima računa se kao zbir poena koje svaka karta koju ima nosi. Broj poena za kartu koja ima vrednost manju od 10 je isti kao i njena vrednost. Broj poena za kartu koja ima vrednost veću ili jednaku 10 je 10. Ukoliko je znak neke karte isti kao i glavni znak u partiji, broj poena se za tu kartu duplira. Program treba da ispiše redni broj igrača (od 1 do N) i broj poena koji je osvojio taj igrač, i to u posebnim redovima, u nerastućem poretku broja poena. Smatrati da je ulaz ispravan.

Primer ulaza:	Primer izlaza:
2 K	2 49
11 K 12 H 2 K 5 T 6 T	1 45
12 P 13 P 2 H 12 K 7 T	

II PITANJA

1) Na nekom računaru, realni brojevi se predstavljaju u formatu **se...em...m**, gde je **s** predznak broja, **e...e** su k bitova eksponenta u kodu sa viškom $2^{k-1} - 1$ i **m...m** su p bitova normalizovane mantise sa skrivenim bitom ($1 \leq M < 2$). Sva zaokruživanja se obavljaju prema pravilima ANSI/IEEE standarda za realne brojeve. Neka se u lokaciju A učitava broj čija je vrednost -332_8 , a u lokaciju B broj čija je vrednost 81.5_{10} . Ukoliko se na datom računaru obavi operacija $C = A + B$, koje su potrebne i dovoljne vrednosti za k i p da bi dobijeni rezultat mogao da se bez zaokruživanja smesti u lokaciju C?

- A) $k=5, p=8$ **(B)** $k=4, p=8$ C) $k=4, p=7$

2) Na jednom računaru, realni brojevi se predstavljaju na širini od 9 bita u formatu **seeeemmmm**, gde je **s** bit predviđen za kodiranje predznaka broja, **eeee** su 4 bita za eksponent u kodu sa viškom 7, a **mmmm** su biti normalizovane matise sa skrivenim bitom ($1 \leq M < 2$). Celi brojevi na ovom računaru se predstavljaju u drugom komplementu na širini od 9 bita. Predstava realnog broja na lokaciji X je $0A4_{16}$. Predstava celog broja na lokaciji Y je $0A4_{16}$. Koja je vrednost realne promenljive Z nakon što se na ovom računaru obavi operacija $Z = X - Y$? Sva zaokruživanja se obavljaju prema pravilima ANSI/IEEE standarda za realne brojeve.

- (A)** -152_{10} B) -154_{10} C) 154.5_{10}

3) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
int main(void) {
    int n = 0, a = 2, b = 3, c = 4, i = n;
    do {
        switch(++i) {
            case 1: n = b;
            case 3: n += c / a * a; break;
            default: n += n&c ? c : -c; continue;
        }
        ++n;
    } while(n >= 2 * i);
    printf("%d %d", n, i);
    return 0;
}
```

- A) 8 6 **(B)** 5 4 C) 12 8

4) Šta ispisuje sledeći program na programskom jeziku C? Smatrati da konverzija `%hhx` ispravno ispisuje elemente niza u heksadecimalnom obliku.

```
#include <stdio.h>
#define CHARBIT 8
void main () {
    unsigned char arr[3] = {0x34, 017}, x, c = 0;
    int i, n = sizeof(arr) / sizeof(unsigned char);
    for (i = 0; i < n; i++) {
        x = arr[i];
        x ^= x >> 4;
        x ^= x >> 2;
        x ^= x >> 1;
        arr[i] |= (x & 1 && ++c) << CHARBIT - 1;
    }
    for (; i > 0; printf("%hhx ", arr[--i]));
    printf("%d\n", c);
}
```

- A) 80 8f b4 3 **(B)** 0 f b4 1 C) 0 8f 34 2

13S111P1, Programiranje 2, prvi kolokvijum 2017/2018. Rešenje zadatka

Zadatak

Napisati program na programskom jeziku C koji računa pobednika u nekoj kartaškoj igri. Igra se igra sa 52 karte. Karta je određena vrednošću i znakom. Vrednost može biti ceo broj od 2 do 14. Postoje četiri znaka čije su jednoslovne oznake K, H, T i P. Korisnik u prvom redu na standardnom ulazu unosi broj igrača N koji igraju igru, kao i glavni znak u partiji. Maksimalan broj igrača je 10. Nakon toga se u N redova, za svakog igrača u posebnom redu, unose karte koje poseduje. Broj karata koje jedan igrač ima je 5. Za svaku kartu se unosi vrednost karte i znak, odvojeni tačno jednim razmakom. Karte u redu su međusobno odvojene jednim razmakom. Broj poena koje jedan igrač ima računa se kao zbir poena koje svaka karta koju ima nosi. Broj poena za kartu koja ima vrednost manju od 10 je isti kao i njena vrednost. Broj poena za kartu koja ima vrednost veću ili jednaku 10 je 10. Ukoliko je znak neke karte isti kao i glavni znak u partiji, broj poena se za tu kartu duplira. Program treba da ispiše redni broj igrača (od 1 do N) i broj poena koji je osvojio taj igrač, i to u posebnim redovima, u nerastućem poretku broja poena. Smatrati da je ulaz ispravan.

```
#include <stdio.h>
#define MAX_NUMBER_OF_PLAYERS 10
#define NUMBER_OF_CARDS 5
#define MAX_CARD_VALUE 10
void main() {
    int numberOfPlayers, players[MAX_NUMBER_OF_PLAYERS],
        points[MAX_NUMBER_OF_PLAYERS], cardValue, i, j, temp;
    char cardSign, mainSign;
    scanf("%d %c", &numberOfPlayers, &mainSign);
    for (i = 0; i < numberOfPlayers; i++) {
        players[i] = i + 1; points[i] = 0;
        for (j = 0; j < NUMBER_OF_CARDS; j++) {
            scanf("%d %c", &cardValue, &cardSign);
            if (cardValue > MAX_CARD_VALUE) {
                cardValue = MAX_CARD_VALUE;
            }
            if (cardSign == mainSign) {
                cardValue *= 2;
            }
            points[i] += cardValue;
        }
    }
    for (i = 0; i < (numberOfPlayers - 1); i++)
        for (j = (i + 1); j < numberOfPlayers; j++) {
            if (points[i] < points[j]) {
                temp = points[i];
                points[i] = points[j];
                points[j] = temp;
                temp = players[i];
                players[i] = players[j];
                players[j] = temp;
            }
        }
    for (i = 0; i < numberOfPlayers; i++)
        printf("%d %d\n", players[i], points[i]);
}
```