

| | | | |
|-------|--|--------------|--|
| JMBAG | | Prezime, ime | |
|-------|--|--------------|--|

Uvod u programiranje

Zimski rok – 8. veljače 2021

Opaske:

- Rješenja moraju biti u skladu sa stilom pisanja preporučenim na predmetu
- Rješenja problema 1 - 4 zapišite na vlastite papire
- Rješenja problema 4 -8 zapišite u dolje predviđeni prostor
- U primjerima znakovi · i ↵ označavaju razmak, odnosno novi redak. Podebljani tekst predstavlja ulaz.

1. (18 bodova)

Napišite funkciju koja briše znak iz niza. Funkcija ne smije koristiti nijedan pomoćni niz. Prototip je `void delete(char *chr)`

gdje je `chr` pokazivač na znak koji se želi izbrisati. `chr` sigurno ne pokazuje na `'\0'`.

U glavnom programu pročitajte niz, pročitajte znak koji želite izbrisati iz niza i pomoću odgovarajuće funkcije iz standardne biblioteke pronađite adresu prvog takvog znaka unutar niza. Ako je znak pronađen, izbrišite ga pomoću funkcije *delete* i ispišite rezultat, u suprotnom ispišite poruku "Ne postoji".

Ulazni niz može sadržavati razmake i sigurno neće imati više od 80 znakova.

Primjeri izvođenja programa:

```
Ulaz· ·>New York↵Y↵
Izlaz·>New ork↵
```

```
Ulaz· ·>Zagreb↵Z↵
Izlaz·>Ne postoji↵
```

2. (18 bodova)

Napišite funkciju koja vraća najmanju pozitivnu vrijednost i najveću pozitivnu vrijednost iz zadanog dvodimenzionalnog cjelobrojnog polja. Ako u polju nema pozitivnih vrijednosti, funkcija vraća *false*, inače *true*.

U glavnom programu pročitajte dimenzije i elemente polja, pozovite funkciju i ispišite rezultate ili tekst "Nema pozitivnih vrijednosti".

Primjeri izvođenja programa:

```
Ulaz· ·>2 2↵1 2↵3 4↵
Izlaz·>1 4↵
```

```
Ulaz· ·>2 2↵0 -1↵-3 -4↵
Izlaz·>Nema pozitivnih vrijednosti↵
```

3. (18 bodova)

Simulirajte jednostavnu igru ruleta. Igrač započinje sa 100 kn, ulaže određeni iznos novca na jedan od brojeva iz intervala [1,36], a računalo okreće rulet. Ako se kuglica zaustavi na 0 ili na bilo kojem broju na koji igrač nije uložio, igrač gubi. Ako je igrač pogodio, dobit će iznos jednak 36 uloga. Nakon svakog uloga ispišite izvučeni broj i trenutni iznos koji igrač posjeduje. Igrač ne može uložiti više novca nego što ima i ne može uložiti na broj izvan intervala [1,36], što se mora biti provjeriti u programu.

Igra završava kad igrač izgubi sav novac (što će se sigurno dogoditi prije ili kasnije).

4. (18 bodova)

Binarna datoteka *f.bin* sadrži nesortirane zapise oblika: šifra (raspon 1000 do 2000, *int*), ime (niz, 40 + 1 znak). U datoteci ne postoje sve šifre iz raspona 1000 do 2000. Stvorite novu datoteku *d.bin* koja će omogućiti izravan pristup imenu preko šifre, bez nepotrebno gubljenja prostora na disku. U isti program unosite šifre i ispisujte odgovarajuća imena, sve dok se ne unese nepostojeća šifra. Nije potrebno provjeravati jesu li ulazno/izlazne operacije uspjele.

Primjer izvođenja programa:

```
Ulaz· ·>200 12↵
Izlaz·>Nedozvoljeno↵
Ulaz· ·>50 0↵
Izlaz·>Nedozvoljeno↵
Ulaz· ·>50 12↵
Izlaz·>33 50↵
Ulaz· ·>10 30↵
Izlaz·>30 410↵
Ulaz· ·>200 10↵
Izlaz·>7 210↵
Ulaz· ·>210 15↵
Izlaz·>16 0↵
```

5. (5 bodova)

Pod pretpostavkom da je varijabla *x* istoga tipa kao rezultat izraza s desne strane operatora pridruživanja u naredbi

```
x = 3 / 2 + 0.125;
```

napišite heksadekadski prikaz sadržaja varijable *x* nakon izvršavanja te naredbe.

Rezultat:

6. (6 bodova)

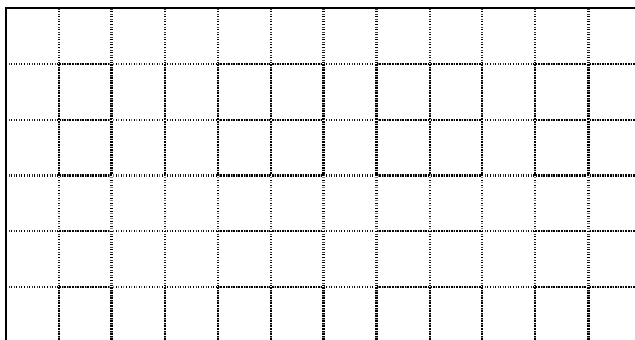
Unesite retke koji nedostaju u funkciji *strcat*:

```
char *strcat(char *s1, const char *s2 ) {  
    char *r;  
  
    _____;  
  
    _____;  
    for (; *s2; s1++, s2++) *s1 = *s2;  
  
    _____;  
    return r;  
}
```

7. (5 bodova)

Pokažite što će se ispisati na ekranu tijekom izvođenja sljedećeg programskog odsječka, uzimajući u obzir točan položaj ispisanih znakova: svaki kvadrat na papiru predstavlja mjesto na ekranu na kojem se ispisuje jedan znak; svaki redak predstavlja jedan redak na zaslonu.

```
int a[] = {32, 50, 68, 65};  
int *p = a + 1;  
printf("d%d%d%c\nn",  
        *a, *p, *(p+2));  
*(p+1) = 10;  
printf("%cc%dd%cc", a[0], a[1], a[2]);  
printf("FER%d", *p + 1);
```



8. (5 bodova)

Što će se ispisati izvođenja sljedećeg programskog odsječka

```
int a = 9, b = 9;  
float c = 9, d = 9;  
int x;  
x = scanf("%4d%d%f%4f", &a, &b, &c, &d);  
printf("%d%d%f%f%d", a, b, c, d, x);
```

ako je na ulazu (• označava prazninu)

••123456.35A6•123456.12

Rezultat:

RJEŠENJA

1.

```
#include <stdio.h>
#include <string.h>
#define MAX 80
void delete (char *chr) {
    do {
        *chr = *(chr + 1);
    } while (*(++chr) != '\0');
}
int main(void) {
    char str[MAX + 1], *p, c;
    scanf("%[^\\n]%*c%c", str, &c);
    if ((p = strchr(str, c)) != NULL) {
        delete (p);
        printf("%s\\n", str);
    } else {
        printf("Ne postoji\\n");
    }
    return 0;
}
```

2.

```
#include <stdbool.h>
#include <stdio.h>
bool minmax(int m, int n, int a[m][n], int *min, int *max) {
    bool ret = false;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            if (a[i][j] > 0) {
                if (!ret) {
                    ret = true;
                    *max = *min = a[i][j];
                } else if (a[i][j] > *max) {
                    *max = a[i][j];
                } else if (a[i][j] < *min) {
                    *min = a[i][j];
                }
            }
        }
    }
    return ret;
}
```

```
int main(void) {
    int m, n, min, max;
    scanf("%d %d", &m, &n);
    int a[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    if (minmax(m, n, a, &min, &max)) {
        printf("%d %d\\n", min, max);
    } else {
        printf("Nema pozitivnih vrijednosti\\n");
    }
    return 0;
}
```

3.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(void) {
    int total = 100, bet, number;
    srand(time(NULL));
    do {
        while (1) {
            scanf("%d %d", &bet, &number);
            if (bet <= total && number >= 1 && number <= 36)
                break;
            printf("Nedozvoljeno\n");
        }
        total -= bet;
        number = rand() % 36 + 1;
        if (number == bet) {
            total += 36 * bet;
        }
        printf("%d %d\n", number, total);
    } while (total > 0);
    return 0;
}
```

4.

```
#include <stdio.h>
#define MINCODE 1000
int main(void) {
    typedef struct {
        int code;
        char name[40 + 1];
    } record_t;
    record_t rec;
    int code;
    FILE *fi, *fo;
    /* priprema testne datoteke (nije dio zadatka)
    FILE *ft;
    record_t test[] = {{1000, "Edgar"}, {1010, "Robert"}, {1020, "Fred"}};
    ft = fopen("f.bin", "wb");
    for (int i = 0; i < sizeof(test) / sizeof(test[0]); i++) {
        fwrite(&test[i], sizeof(test[i]), 1, ft);
    }
    fclose(ft);
    */
    fi = fopen("f.bin", "rb");
    fo = fopen("d.bin", "wb");
    while (fread(&rec, sizeof(rec), 1, fi) == 1) {
        fseek(fo, (rec.code - MINCODE) * sizeof(rec), SEEK_SET);
        fwrite(&rec, sizeof(rec), 1, fo);
    }
    while (1) {
        scanf("%d", &code);
        fseek(fo, (code - MINCODE) * sizeof(rec), SEEK_SET);
        if (fread(&rec, sizeof(rec), 1, fo) != 1 || rec.code != code)
            break;
        printf("%s\n", rec.name);
    }
    return 0;
}
```

5.

3FF2000000000000

6.

```
char *strcat(char *s1, const char *s2 ) {  
    char *r;  
    r = s1;  
    for (; *s1; s1++);  
    for (; *s2; s1++, s2++) *s1 = *s2;  
    *s1 = '\\0';  
    return r;  
}
```

7.

| | | | | | | | | | | | |
|---|---|---|---|---|---|--|--|--|--|--|--|
| d | 3 | 2 | 5 | 0 | A | | | | | | |
| n | | c | 5 | 0 | d | | | | | | |
| c | F | E | R | 5 | 1 | | | | | | |

8.

1234560.3500009.0000003