

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

Uvod u programiranje

1. ispitni rok - 10. veljače 2020.

Rješenja zadataka 1 - 5 napisati na vlastitim papirima i predati ih u košuljici.

Rješenja zadataka 6 - 10 napisati na za to predviđeno mjesto uz tekst zadatka te listove sa zadacima predati u košuljici.

1. (12 bodova)

Svaki zapis *tekstne* datoteke `stud.txt` sadrži podatke o jednom studentu. Zapis sadrži matični broj (cijeli broj) i datum rođenja studenta (tri cijela broja međusobno odijeljena točkama).

Svaki zapis *binarne* datoteke `bodovi.bin` sadrži matični broj studenta (*int*), dvoranu u kojoj je pisao ispit (niz znakova, *string*, najveće moguće duljine 10+1 znak) i broj bodova koje je dobio na ispitu (*float*). Redni broj zapisa u datoteci `bodovi.bin` odgovara matičnom broju studenta.

Napisati program kojim će se studentima navedenim u datoteci `stud.txt` i koji ispit nisu pisali u dvorani D259, broj bodova u datoteci `bodovi.bin` postaviti na prosječni broj bodova. Prosječni broj bodova je aritmetička sredina svih bodova u datoteci `bodovi.bin`.

Primjer sadržaja `stud.txt`

```
550·19.3.2002.␣
343·23.2.2003.␣
12·1.12.2004.␣
13·17.8.2001.␣
...
```

Primjer sadržaja `bodovi.bin`

...
11	D259	12.0
12	B2	17.6
13	D259	14.7
...
15	PCLAB2	17.3
...
343	D270	15.7
...
345	A102	9.4
346	A101	10.0
...
550	D259	16.9
...

2. (12 bodova)

Napisati funkciju `slucajni` koja za zadani niz znakova (*string*) koji sigurno sadrži barem jedan znak, vraća pokazivač na slučajno odabrani znak u nizu. Generator pseudo-slučajnih brojeva treba inicijalizirati u funkciji.

U nastavku napisati funkciju `main` ("glavni program") koja će pomoću dva uzastopna poziva funkcije `slucajni` dobiti dva pokazivača na znakove u konstantnom znakovnom nizu "Krivotvoritelji·diploma", te na temelju tih pokazivača ispisati sve znakove koji se nalaze između znakova na koje oni pokazuju, odnosno poruku "Nema·znakova" ako između znakova na koje pokazivači pokazuju nema niti jedan znak.

Primjeri izvršavanja programa:

Ako su vraćeni pokazivač na prvo slovo o i pokazivač na treće slovo o

Ako su vraćeni pokazivač na treće slovo o i pokazivač na prvo slovo o

Ako su vraćeni pokazivač na znak m i pokazivač na znak a

Ako su vraćeni pokazivač na znak m i pokazivač na znak m

tvoritelji·dipl

Nema·znakova

Nema·znakova

Nema·znakova

3. (12 bodova)

- Napisati potpuni sadržaj modula koji sadrži funkciju `jestPrimBroj` koja za zadani cijeli broj vraća logičku vrijednost *istina* ili *laž* ovisno o tome je li preko parametra zadani cijeli broj (tipa *int*) prim broj.
- Napisati potpuni sadržaj modula koji sadrži funkciju `main` ("glavni program"). U glavnom programu s tipkovnice uzastopno učitavati cijele brojeve, za svaki učitani broj pozivom funkcije `jestPrimBroj` testirati je li prim broj. Ako broj jest prim broj, ispisati poruku "Jest prim broj.␣", u suprotnom ispisati poruku "Nije prim broj.␣" i prekinuti program.
- Napisati naredbe za zasebno prevođenje (*compiling*) modula iz a) i modula iz b) dijela zadatka, te naredbu za povezivanje (*linking*) rezultata dobivenih prevođenjem u izvršni programski kod.

Napomena: Jasno naznačiti imena i sadržaj datoteka koje čine modul pod a) i modul pod b)

4. (12 bodova)

Članovi jednodimenzijskog polja su strukture koje sadrže x i y koordinate točke (koordinate su tipa *double*). Napisati funkciju `transponiraj` koja će svim točkama u jednodimenzijskom polju zamijeniti njihove vrijednosti x i y koordinata, a u pozivajući program vratiti dvije vrijednosti (ne strukturu koja sadrži te dvije vrijednosti): broj točaka kojima su obje koordinate pozitivne i broj točaka kojima su obje koordinate negativne.

5. (12 bodova)

Napisati funkciju kvadmat koja u zadanoj cjelobrojnoj kvadratnoj matrici na vrijednost nula postavlja najveće i najmanje članove u svakom retku. Npr. u matrici na slici, funkcija će na vrijednost nula postaviti članove označene kružnicama.

2	1	5	9
-9	5	-9	5
7	2	2	4
0	2	3	6

6. (4 boda)

Na predviđena mjesta napisati vrijednost i tip rezultata sljedećeg aritmetičkog izraza:

$(11 \ \& \ 7 \ \<< \ 3 \ \>> \ 2) + 5. \ / \ -2 + 5 * 3 \ / \ 2$

vrijednost	tip
------------	-----

7. (4 boda)

U skladu sa standardom IEEE 754 za prikaz brojeva standardne preciznosti, u registru je pohranjen negativni realni broj $-3.25 \cdot 2^{-127}$. Sadržaj tog registra prikažite u **oktalnom** brojevnom sustavu.

8. (4 boda)

Prikazati što će se tijekom izvršavanja odsječka programa ispisati na zaslonu, vodeći računa o točnoj poziciji ispisanih znakova: svaki "kvadratić" na papiru predstavlja mjesto na zaslonu na koje se ispisuje po jedan znak; svaki "redak kvadratića" predstavlja jedan redak na zaslonu.


```
int i = 10, j = 20;
static int a, b, c;
a = ++b == 1 || c-- == -1;
{
    int i = 15; j = 25;
}
printf("%d %d\n", i, j);
printf("%d %d %d", a, b, c);
```

9. (4 boda)

Prikazati što će se tijekom izvršavanja odsječka programa ispisati na zaslonu, vodeći računa o točnoj poziciji ispisanih znakova: svaki "kvadratić" na papiru predstavlja mjesto na zaslonu na koje se ispisuje po jedan znak; svaki "redak kvadratića" predstavlja jedan redak na zaslonu. Pretpostaviti da je preko tipkovnice upisan sadržaj prikazan u lijevom okviru.

```
..50...┘
-1.25·A1.1 22.21┘
```

```
int a = 0, b = 0;
float x = 5.5, y = 2.5;
b = scanf("%d%f%f", &a, &x, &y);
printf("%d.%c\n", a, a + 1);
printf("%d\n", b);
printf("%.2f%.2f\n", x, y);
printf("%d", sizeof(stdin));
```


10. (4 boda)

Koja od funkcija fact1 i fact2 pri izračunavanju faktoriijela za n=5 koristi više memorije i zašto?

```
int fact1(int n) {
    int rez;
    if (n == 0)
        return 1;
    else
        rez = n * fact(n - 1);
    return rez;
}
```

```
int fact2(int n) {
    int rez = 1;
    for (int i = 2; i <= n; ++i)
        rez = rez * i;
    return rez;
}
```

RJEŠENJA

```
1. #include <stdio.h>
#include <string.h>

int main(void) {
    struct bodovi_s {
        int mbrStud;
        char dvorana[10];
        float bodova;
    } bodZapis;
    FILE *txt = fopen("studenti.txt", "r");
    FILE *bin = fopen("bodovi.bin", "r+b");
    float sumaBodova = 0.f;
    int brojZapisa = 0;
    while (fread(&bodZapis, sizeof(bodZapis), 1, bin) == 1) {
        if (bodZapis.mbrStud != 0) {
            sumaBodova += bodZapis.bodova;
            ++brojZapisa;
        }
    }
    float prosjBodova = sumaBodova / brojZapisa;
    int mbrStud;
    while (fscanf(txt, "%d%d.%d.%d", &mbrStud) == 1) {
        fseek(bin, (long)(mbrStud - 1) * sizeof(bodZapis), SEEK_SET);
        fread(&bodZapis, sizeof(bodZapis), 1, bin);
        if (strcmp(bodZapis.dvorana, "D259") != 0) {
            bodZapis.bodova = prosjBodova;
            fseek(bin, -1L * sizeof(bodZapis), SEEK_CUR);
            fwrite(&bodZapis, sizeof(bodZapis), 1, bin);
        }
    }
    fclose(txt);
    fclose(bin);

    return 0;
}

2. #include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
char *slucajni(char *niz) {
    static int prviPoziv = 1;
    if (prviPoziv) {
        srand((unsigned int)time(NULL));
        prviPoziv = 0;
    }
    return niz + rand() % strlen(niz);
}

int main(void) {
    char *niz = "Krivotvoritelji diploma";
    char *p1 = slucajni(niz);
    char *p2 = slucajni(niz);
    if (p2 >= p1 + 2) {
        for (; p1 < p2; ++p1) {
            printf("%c", *p1);
        }
    } else {
        printf("Nema znakova");
    }
    return 0;
}
```

3. a)

z3m1.c

```
#include "z3m1.h"
_Bool jestPrimBroj(int n) {
    _Bool djeljiv = 0;
    for (int i = 2; i <= n - 1; i = i + 1) {
        if (n % i == 0) {
            djeljiv = 1;
            break;
        }
    }
    return n > 1 && !djeljiv;
}
```

z3m1.h

```
_Bool jestPrimBroj(int n);
```

b)

z3m2.c

```
#include <stdio.h>
#include "z3m1.h"
int main(void) {
    int n;
    _Bool jestPrim;
    do {
        scanf("%d", &n);
        jestPrim = jestPrimBroj(n);
        if (jestPrim) {
            printf("Jest prim broj\n");
        } else {
            printf("Nije prim broj\n");
        }
    } while (!jestPrim);
    return 0;
}
```

c) gcc -std=c11 -Wall -pedantic-errors -c z3m1.c
gcc -std=c11 -Wall -pedantic-errors -c z3m2.c
gcc -o z3.exe z3m1.o z3m2.o

4. struct tocka_s {

```
    double x;
    double y;
```

```
};
```

```
void transponiraj(struct tocka_s *polje, int n, int *objePozitivne, int *objeNegativne) {
    *objePozitivne = *objeNegativne = 0;
    for (int i = 0; i < n; ++i) {
        if ((polje + i)->x < 0. && (polje + i)->y < 0.) {
            ++*objeNegativne;
        } else if ((polje + i)->x > 0. && (polje + i)->y > 0.) {
            ++*objePozitivne;
        }
        double pom = (polje + i)->x;
        (polje + i)->x = (polje + i)->y;
        (polje + i)->y = pom;
    }
    return;
}
```

```

5. void kvadmat(int *mat, int n) {
    for (int i = 0; i < n; ++i) {
        int najveci, najmanji;
        najveci = najmanji = *(mat + n * i + 0);
        for (int j = 1; j < n; ++j) {
            if (*(mat + n * i + j) > najveci) {
                najveci = *(mat + n * i + j);
            }
            if (*(mat + n * i + j) < najmanji) {
                najmanji = *(mat + n * i + j);
            }
        }
        for (int j = 0; j < n; ++j) {
            if (*(mat + n * i + j) == najveci || *(mat + n * i + j) == najmanji) {
                *(mat + n * i + j) = 0;
            }
        }
    }
    return;
}

```

6.

14.5	vrijednost
------	------------

double	tip
--------	-----

7.

2 0 1 1 2 0 0 0 0 0 0	prostor za odgovor
-----------------------	--------------------

8.

1	0	2	5	
1		1	0	

9.

5	0		3							
2										
-	1	.	2	5			2	.	5	0
4										

10. Funkcija fact1.
Pri svakom rekurzivnom pozivu na stog mora staviti argument, povratnu adresu i lokalnu varijablu.