

1. Proučite, prevedite i testirajte sljedeći program. Navedite gdje se u programu nalaze:

- [1-4] a) Direktive pretprocesora [na vrhu (`#include <stdio.h>`)]
 7,8,9 b) Definicije varijabli [drugi red nakon main (podgrana)]
 [2-4] c) Definicije simboličkih konstanti [ispod direktiva, iznad imenovanog bloka]
 2,3,4,14,15 d) Aritmetički izrazi [unutar imenovanog bloka]
 14,15 e) Izrazi pridruživanja [na kraju imenovanog bloka]

Zatim editorom pregledajte rezultat pretprocesiranja izvornog programskog koda. Što je pretprocesor obavio u vezi naredbe `include`, što u vezi naredbi `define`, a što je učinio s komentarima u izvornom kodu?

rež. procesiranja : `gcc -E prog.c > prog.i`
 notepad prog.i

```
1 #include <stdio.h>
2 #define MASA_ZEMLJE 5.9722e24f // kg
3 #define POLUMJER_ZEMLJE 6.371e6f // m
4 #define GRAV_KONST 6.67408e-11f // m(3) kg(-1) s(-2)
5
6 int main(void) {
7     int h; // visina iznad površine Zemlje
8     float g;
9     float udalj_sr; // ukupna udaljenost do sredista Zemlje
10
11     printf("Upisite visinu (m) iznad površine Zemlje > ");
12     scanf("%d", &h);
13
14     udalj_sr = POLUMJER_ZEMLJE + h;
15     g = MASA_ZEMLJE * GRAV_KONST / (udalj_sr * udalj_sr);
16
17     printf("gravitacija na visini %d metara je približno %e m s(-2)", h, g);
18
19     return 0;
20 }
```

2. Napisati program koji će s tipkovnice učitati realni broj veći od nule koji predstavlja radijus kugle. Ako je radijus ispravno upisan, ispisati radijus i volumen kugle. Inače, ispisati poruku *Neispravan radijus kugle*.

Primjeri izvršavanja programa

Upisite radijus kugle > 1.1
 Volumen kugle radijusa 1.000 je 4.189

Upisite radijus kugle > 50.15
 Volumen kugle radijusa 50.150 je 528324.938

Upisite radijus kugle > -1.1
 Neispravan radijus kugle

zasto .f dodavati brojima
 + kako se koristi
 + zasto kada matrem
 .f sa 4/3 dobijem
 krivi broj?

3. Napisati program koji će s tipkovnice učitati dva cijela broja. Ako je jedan od učitanih brojeva veći od 100, a onaj drugi manji od 100, ispisati poruku *Jedan je veci, drugi je manji od 100* (uočiti: piše jedan od učitanih brojeva, a ne prvi učitani broj). Inače, ne ispisati ništa.

4. Napisati program za pretvaranje vrijednosti temperature izražene u različitim mjernim jedinicama. Na početku program treba ispisati poruku:

Program za konverziju Fahrenheit - Celsius ili obrnuto.

Za F u C upisite 1, a za C u F bilo koji drugi cijeli broj >

Ako korisnik za izbor smjera konverzije upiše broj 1, tada ispisati poruku

Upisite temperaturu izraženu u stupnjevima Fahrenheit >

te s tipkovnice učitati realni broj koji predstavlja temperaturu izraženu u stupnjevima Fahrenheit, izračunati ekvivalentnu temperaturu izraženu u stupnjevima Celsius te ispisati temperaturu izraženu u obje jedinice.

Ako korisnik za izbor smjera konverzije upiše bilo koji drugi cijeli broj, tada ispisati poruku

Upisite temperaturu izraženu u stupnjevima Celsius >

te s tipkovnice učitati jedan realni broj koji predstavlja temperaturu izraženu u stupnjevima Celsius, izračunati ekvivalentnu temperaturu izraženu u stupnjevima Fahrenheit te ispisati temperaturu izraženu u obje jedinice.

Primjeri izvršavanja programa

Program za konverziju Fahrenheit - Celsius ili obrnuto.

Za F u C upisite 1, a za C u F bilo koji drugi cijeli broj > **1**

Upisite temperaturu izraženu u stupnjevima Fahrenheit > **451**

451.000 st. F = 232.778 st. C

Program za konverziju Fahrenheit - Celsius ili obrnuto.

Za F u C upisite 1, a za C u F bilo koji drugi cijeli broj > **5**

Upisite temperaturu izraženu u stupnjevima Celsius > **20.25**

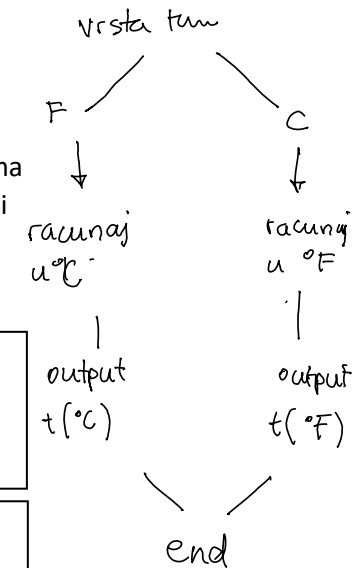
20.250 st. C = 68.450 st. F

$$t(^{\circ}\text{C}) \cdot \frac{9}{5} = t(^{\circ}\text{F}) - 32$$

$$\frac{9}{5}t(^{\circ}\text{C}) + 32 = t(^{\circ}\text{F})$$

$$(32^{\circ}\text{F} - 32) \times \frac{5}{9} = 0^{\circ}\text{C}$$

$$\frac{5}{9}(t(^{\circ}\text{F}) - 32) = t(^{\circ}\text{C})$$



- Napisati program koji će s tipkovnice učitati dva cijela broja x i y. Ako je x barem za 100 veći od y i pri tome je y ili paran ili negativan, ispisati poruku *Istina je*. Inače, ne ispisati ništa.
- Napisati program koji će ispisati poruku *Istina je* ako uvjet iz prethodnog zadatka nije zadovoljen. Jedno rješenje napisati izravnom primjenom operatora negacije na uvjetu iz rješenja prethodnog zadatka. Drugo rješenje napisati bez operatora negacije (preoblikovanje logičkog izraza obaviti pomoću De Morganovog pravila).
- Napisati program koji će s tipkovnice učitati dva realna broja x i y. Ispisati poruku *Istina je* ako je barem jedan od brojeva u zatvorenom intervalu [5, 20] i pri tome je x manji od y. Inače, ispisati poruku *Nije istina*.
- Napisati program koji će s tipkovnice učitati tri cijela broja. Ako su učitane vrijednosti međusobno različite i poredane od manjih prema većim ili od većih prema manjim (npr. za učitane brojeve -5, 1, 18 ili 54, 10, 1), ispisati poruku *Brojevi su poredani i razliciti*, inače (npr. za učitane brojeve 5, 2, 6 ili 4, 4, 5) ispisati poruku *Brojevi nisu poredani ili nisu razliciti*.
- Napisati program koji će s tipkovnice učitati pet realnih brojeva. Ako su barem tri od učitanih brojeva veći od nule, ispisati poruku *Barem tri su veka od nule*, inače ispisati poruku *Niti tri nisu veka od nule*.

10. Proučite i zaključite što bi sljedeći program (kada bi bio ispravan) trebao raditi.

main
int
scanf

```
#include <stdio.h>
int main(void) {
    int xA, yA, xB, yB, xC, yC;

    printf("Upisite koordinate vrha trokuta A > ");
    scanf("%d %d", &xA, &yA);
    printf("Upisite koordinate vrha trokuta B > ");
    scanf("%d %d", &xB, &yB);
    printf("Upisite koordinate vrha trokuta C > ");
    scanf("%d %d", &xC, &yC);

    xTezista = (xA + xB + xC) / 3;
    yTezista = (yA + yB + yC) / 3;

    printf("Teziste trokuta A(%d, %d), B(%d, %d), C(%d, %d) jest T(%d, %d)",
           xA, yA, xB, yB, xC, yC, xTezista, yTezista);
    return 0;
}
```

ne
stvari
trokut → *Edi* x Tezista
y Tezista
→ *inače samo xC(yC) dijeli*
→ *ovo treba isti red ili dvobitku*

Zatim pomoću *copy-paste* upišite program u datoteku na svom računalu i pokušajte ga prevesti. Proučite poruke koje je dojavio prevodilac. Prvo ispravite isključivo sintaktičke (formalne) pogreške. Ponovo pokušajte prevesti program i proučite poruke koje je dojavio prevodilac. Ispravite pogreške povezivanja. Kada se program uspješno prevede, testirajte program s različitim ulaznim podacima:

- za vrhove trokuta s koordinatama (0, 0), (3, 3) i (6, 0)
- za vrhove trokuta s koordinatama (0, 0), (1, 1) i (2, 0)
- za vrhove trokuta s koordinatama (0, 0), (1.5, 1.5) i (3, 0)

Ispravite sve logičke pogreške. Po čemu bitnom se logičke pogreške razlikuju od sintaktičkih pogrešaka i pogrešaka povezivanja?

ode me se mozaik a ne samo za oči

11. U programu definirati sljedeće varijable i pridružiti im vrijednosti (kopirati, *copy-paste*):

```
float napierConst = 2.71828f;
float saturnSunDist = 1.43344e12f;
float planckConst = 6.62607e-34f;
```

Zatim, u nastavku programa, ispisati njihove vrijednosti točno kako je prikazano u primjeru izvršavanja programa.

Primjer izvršavanja programa

The screenshot shows the output of a C program. Handwritten annotations in blue and black ink are present:

- A blue bracket on the left groups the first three lines of output (Napier's constant) and is labeled with a blue '6'.
- A blue arrow points from the text 'Ove točke označavaju praznine (blank, space). U zadacima će se koristiti u svim slučajevima u kojima programerima treba olakšati brojanje praznina ("koliko praznina trebam ispisati na ovom mjestu?").' to the spaces in the output.
- A blue bracket on the right groups the last three lines of output (Planck's constant) and is labeled with a blue '6'.
- A blue bracket on the left groups the middle two lines of output (Saturn's distance) and is labeled with a blue '4'.
- Black handwritten notes on the right provide instructions: 'Npr. na ovom mjestu treba ispisati dvije praznine, što znači da program treba ispisati realni broj u ukupnoj širini 6 znakova, od toga dvije znamenke iza decimalne točke.' and 'Npr. na ovom mjestu treba ispisati dvije praznine, što znači da program treba ispisati realni broj u ukupnoj širini 6 znakova, od toga dvije znamenke iza decimalne točke.'
- Black handwritten notes on the left indicate the number of digits after the decimal point for each constant: '2' for Napier's constant, '12' for Saturn's distance, and '34' for Planck's constant.

printf
... ("ln %.f", napierConst)
... ("ln %.2f", saturnSunDist)
... ("ln %.10f", planckConst)

ne mogu s petljama :/

printf("ln")

Rješenja:

1. a) Direktive pretprocesoru

```
#include <stdio.h>
#define MASA_ZEMLJE      5.9722e24
#define POLUMJER_ZEMLJE  6.371e6
#define GRAV_KONST       6.67408e-11
```

b) Definicije varijabli

```
int h;
float g;
float udalj_sr;
```

c) Definicije simboličkih konstanti

```
#define MASA_ZEMLJE      5.9722e24
#define POLUMJER_ZEMLJE  6.371e6
#define GRAV_KONST       6.67408e-11
```

d) Aritmetički izrazi (operandi u izrazu su označeni žutom, operatori zelenom bojom)

```
POLUMJER_ZEMLJE + h
udalj_sr * udalj_sr
MASA_ZEMLJE / (udalj_sr * udalj_sr)
MASA_ZEMLJE / (udalj_sr * udalj_sr) * GRAV_KONST
```

e) Izrazi pridruživanja (operandi u izrazu su označeni žutom, operatori zelenom bojom)

```
udalj_sr = POLUMJER_ZEMLJE + h
g = MASA_ZEMLJE / (udalj_sr * udalj_sr) * GRAV_KONST
```

Uz pretpostavku da je izvorni program pohranjen u datoteci prog.c, rezultat pretprocesiranja se najlakše može pogledati ovako:

```
gcc -E prog.c > prog.i
notepad prog.i
```

2. #include <math.h>
#include <stdio.h>
#define PI 3.14159f

```
int main(void) {
    float radijus, volumen;
    printf("Upisite radijus kugle > ");
    scanf("%f", &radijus);

    if (radijus > 0.f) {
        volumen = 4.f / 3 * PI * pow(radijus, 3.f);
        printf("Volumen kugle radijusa %.3f je %.3f", radijus, volumen);
    } else {
        printf("Neispravan radijus kugle");
    }

    return 0;
}
```

3. #include <stdio.h>

```
int main(void) {
    int a, b;
    printf("Upisite dva broja > ");
    scanf("%d %d", &a, &b);

    if ((a > 100 && b < 100) || (a < 100 && b > 100)) {
        printf("Jedan je veci, drugi je manji od 100");
    }

    return 0;
}
```

4. #include <stdio.h>

```
int main(void) {
    float cels, fahr;
    int izbor;

    printf("Program za konverziju Fahrenheit - Celsius ili obrnuto.\n");
    printf("Za F u C upisite 1, a za C u F bilo koji drugi cijeli broj > ");
    scanf("%d", &izbor);

    if (izbor == 1) {
        printf("Upisite temperaturu izrazenu u stupnjevima Fahrenheit > ");
        scanf("%f", &fahr);
        cels = (fahr - 32.f) * 5.f / 9.f;
        printf("%8.3f st. F = %8.3f st. C\n", fahr, cels);
    } else {
        printf("Upisite temperaturu izrazenu u stupnjevima Celsius > ");
        scanf("%f", &cels);
        fahr = cels * 9.f / 5.f + 32.f;
        printf("%8.3f st. C = %8.3f st. F\n", cels, fahr);
    }

    return 0;
}
```

5. #include <stdio.h>

```
int main(void) {
    int x, y;
    printf("Upisite dva broja > ");
    scanf("%d %d", &x, &y);

    if (x >= y + 100 && (y % 2 == 0 || y < 0)) {
        printf("Istina je");
    }

    return 0;
}
```

6. #include <stdio.h>

```

int main(void) {
    int x, y;
    printf("Upisite dva broja > ");
    scanf("%d %d", &x, &y);

    if ((x >= y + 100 && (y % 2 == 0 || y < 0))) {
        printf("Istina je");
    }

    return 0;
}

```

Primjenom De Morganovog pravila

$$\begin{aligned}
 &!(x \geq y + 100 \ \&\& \ (y \% 2 == 0 \ || \ y < 0)) \equiv \\
 &!(x \geq y + 100) \ || \ !(y \% 2 == 0 \ || \ y < 0) \equiv \\
 &(x < y + 100) \ || \ (!(y \% 2 == 0) \ \&\& \ !(y < 0)) \equiv \\
 &(x < y + 100) \ || \ ((y \% 2 != 0) \ \&\& \ (y \geq 0)) \equiv \\
 &(x < y + 100) \ || \ (y \% 2 != 0 \ \&\& \ y \geq 0)
 \end{aligned}$$

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

```

int main(void) {
    int x, y;
    printf("Upisite dva broja > ");
    scanf("%d %d", &x, &y);

    if ((x < y + 100) || (y % 2 != 0 && y >= 0)) {
        printf("Istina je");
    }

    return 0;
}

```

7. #include <stdio.h>

```

int main(void) {
    float x, y;
    printf("Upisite dva broja > ");
    scanf("%f %f", &x, &y);

    if (((x >= 5.f && x <= 20.f) || (y >= 5.f && y <= 20.f)) && x < y) {
        printf("Istina je");
    } else {
        printf("Nije istina");
    }

    return 0;
}

```

8. #include <stdio.h>

```
int main(void) {
    int a, b, c;
    printf("Upisite tri broja > ");
    scanf("%d %d %d", &a, &b, &c);

    if ((a < b && b < c) || (a > b && b > c)) {
        printf("Brojevi su poredani i razliciti");
    } else {
        printf("Brojevi nisu poredani ili nisu razliciti");
    }

    return 0;
}
```

9. #include <stdio.h>

```
int main(void) {
    float b1, b2, b3, b4, b5;
    int brojac = 0;
    printf("Upisite pet brojeva > ");
    scanf("%f %f %f %f %f", &b1, &b2, &b3, &b4, &b5);

    if (b1 > 0.f)
        brojac = brojac + 1;
    if (b2 > 0.f)
        brojac = brojac + 1;
    if (b3 > 0.f)
        brojac = brojac + 1;
    if (b4 > 0.f)
        brojac = brojac + 1;
    if (b5 > 0.f)
        brojac = brojac + 1;

    if (brojac >= 3) {
        printf("Barem tri su veca od nule");
    } else {
        printf("Niti tri nisu veca od nule");
    }

    return 0;
}
```

Napomena: Ovaj zadatak se može elegantnije riješiti pomoću jednodimenzijskog polja, ali dok se na predavanjima ne obrade agregatni tipovi podataka, ovdje priloženo rješenje je prihvatljivo.

10. Sintaktičke pogreške:

- točka-zarez se ne smije stavljati iza pretprocesorskih naredbi
- definicija varijabli x_a , y_a , x_b , y_b , itd. je neispravna. Varijable koje se odjednom definiraju međusobno se odjeljuju zarezom, a ne točkom-zarezom. Tip integer ne postoji
- varijable $x_{Tezista}$ i $y_{Tezista}$ nisu definirane
- nedostaje točka-zarez iza naredbe return

Pogreške povezivanja:

- u `stdio.h` ne postoji funkcija `scan` (postoji `scanf`)
- program mora sadržavati funkciju `main` (a ne `mein`)

Logičke pogreške:

- koordinate težišta se neispravno izračunavaju. X koordinate triju vrhova treba prvo zbrojiti, a tek tada podijeliti s 3 (trenutno se na kvocijent $z/3$ dodaje vrijednost za x i y . Isto vrijedi i za y koordinate.
- nakon dodavanja zagrada program će ispravno raditi za koordinate vrhova (0, 0), (3, 3) i (6, 0), ali ne i za koordinate (0, 0), (1, 1) i (2, 0). Razlog je u cjelobrojnomo dijeljenju i tipu rezultata: $0+1+0$ cjelobrojno podijeljeno s 3 daje 0, a ne 0.33333. Potrebno je promijeniti tip svih varijabli: koordinate vrhova trokuta i težišta trebaju biti realni, a ne cijeli brojevi.

Program nakon ispravljenih pogreški:

```
#include <stdio.h>

int main(void) {
    float  $x_a$ ,  $y_a$ ,  $x_b$ ,  $y_b$ ,  $x_c$ ,  $y_c$ ;
    float  $x_{Tezista}$ ,  $y_{Tezista}$ ;

    printf("Upisite koordinate vrha trokuta A > ");
    scanf("%f %f", & $x_a$ , & $y_a$ );
    printf("Upisite koordinate vrha trokuta B > ");
    scanf("%f %f", & $x_b$ , & $y_b$ );
    printf("Upisite koordinate vrha trokuta C > ");
    scanf("%f %f", & $x_c$ , & $y_c$ );

     $x_{Tezista}$  = ( $x_a$  +  $x_b$  +  $x_c$ ) / 3;
     $y_{Tezista}$  = ( $y_a$  +  $y_b$  +  $y_c$ ) / 3;

    printf("Teziste trokuta A(%f, %f), B(%f, %f), C(%f, %f) jest T(%f, %f)",  $x_a$ ,
         $y_a$ ,  $x_b$ ,  $y_b$ ,  $x_c$ ,  $y_c$ ,  $x_{Tezista}$ ,  $y_{Tezista}$ );

    return 0;
}
```

11. #include <stdio.h>

```
int main(void) {
    float napierConst = 2.71828f;
    float saturnSunDist = 1.43344e12f;
    float planckConst = 6.62607e-34f;

    printf("Napierova konstanta:\n");
    printf("%f\n", napierConst);
    printf("%.2f\n", napierConst);
    printf("%.3f\n", napierConst);
    printf("%.14f\n\n", napierConst);

    printf("Srednja udaljenost Saturna i Sunca:\n");
    printf("%f\n", saturnSunDist);
    printf("%.2f\n", saturnSunDist);
    printf("%.12e\n\n", saturnSunDist);

    printf("Planckova konstanta:\n");
    printf("%f\n", planckConst);
    printf("%.40f\n", planckConst);
    printf("%.5e\n", planckConst);
    printf("%.19e\n", planckConst);

    return 0;
}
```