

Tema 1

Uvod, Rešavanje problema, osnovni
tipovi, standardni ulaz i izlaz

Osnovne informacije

- [Sajt katedre](#)
 - Obaveštenja
 - Rezultati testova
 - Materijali za predavanja i vežbe
 - Sekcija Repozitorijum
- [Praktikum](#)
- [Zbirka zadataka](#)

Platforma za održavanje nastave

- Operativni sistem Linux
 - Distribucija Ubuntu
- Kompajler gcc
- Tekstualni editor code (Visual Studio Code)
- Konzolno okruženje terminal
 - Korišćenje osnovnih komandi za rad sa fajlsistemom

Načini instalacije Linuksa

- Aktuelna verzija sistema u laboratoriji je Ubuntu 20.04
- Virtuelna mašina
 - Preporučen način rada, dovoljan za potrebe predmeta
 - Virtual Box (dva načina instalacije)
 - Pripremljen OVA fajl (dovoljan je import u Virtual Box) na sledećem [linku](#)
 - Ručno nameštanje virtuelne mašine i instalacija Ubuntu-a uz [video uputstvo](#)
- Pored trenutnog operativnog sistema (dual boot)

Windows alternativa

- Podešavanje Terminala i kompajlera pomoću uputstva [na linku](#)
- Za avanturiste, postoji varijanta i preko [Windows Subsystem for Linux \(WSL\)](#)
- Biranje bash varijante prilikom otvaranja Terminala u okviru editora

Napomena:

Ovakav način rada nije zvanično podržan na predmetu. Kolokvijumi na ovom predmetu se isključivo rade na računarima u laboratoriji koji koriste Linux, samim tim se očekuje da student poznaje to okruženje. Instaliranje i pokretanje okruženja na ovaj način radite na sopstvenu odgovornost!

Komandna linija (Terminal)

- Drugačiji pogled na fajlsistem u odnosu na grafički interfejs
- Pojam home direktorijuma
- Osnovne komande
 - Ispisivanje pune putanje trenutnog direktorijuma `pwd`
 - Listanje sadržaja `ls`
 - Pravljenje novog direktorijuma `mkdir`
 - Promena direktorijuma `cd`
 - Pravljenje praznog fajla `touch`
- Poziv editora [Visual Studio Code](#) iz komandne linije
 - Kucanjem komande `code`

Upoznavanje sa okruženjem

1. Otvoriti Terminal
2. Isprobati komande `pwd` i `ls`, analizirati njihov ispis
3. Napraviti novi direktorijum pomoću komande `mkdir pjisp`
4. Ponovo pokrenuti komandu `ls`. Utvrditi da se pojavio direktorijum `pjisp`.
5. Ući u direktorijum pomoću komande `cd pjisp`
6. Otvoriti VS Code pomoću komande `code .` (izabrati *Yes, I trust the authors*)
7. Otvoriti Terminal u okviru editora (Terminal -> New Terminal)

Rešavanje problema

- Razumevanje problema
- Izgradnja modela
- Formulisanje algoritma
- Provera ispravnosti
- Realizacija algoritma (pisanje programa)
- Testiranje programa
- Sastavljanje dokumentacije

Zadatak 1

Izračunati površinu trougla upotrebom Heronovog obrasca.

- $a = 2\text{cm}$
- $b = 4\text{cm}$
- $c = 5\text{cm}$

Razumevanje problema

- Terminološki, šta je Heronov obrazac?
- Šta je dato? Stranice trougla, a , b i c .
- Šta se traži? Površina trougla, P .
- Dopuštene metode. Funkcija za traženje kvadratnog korena (`sqrt`)
- Podaci: $a = 2\text{cm}$, $b = 4\text{cm}$ i $c = 5\text{cm}$

Izgradnja modela

$$s = \frac{a+b+c}{2}$$

$$P = \sqrt{s * (s - a) * (s - b) * (s - c)}$$

Realizacija algoritma

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

int main()
{
    float a = 2.0;
    float b = 4.0;
    float c = 5.0;
    float s, P;
    s = (a + b + c) / 2;
    P = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("P = %f\n", P);

    return 0;
}
```

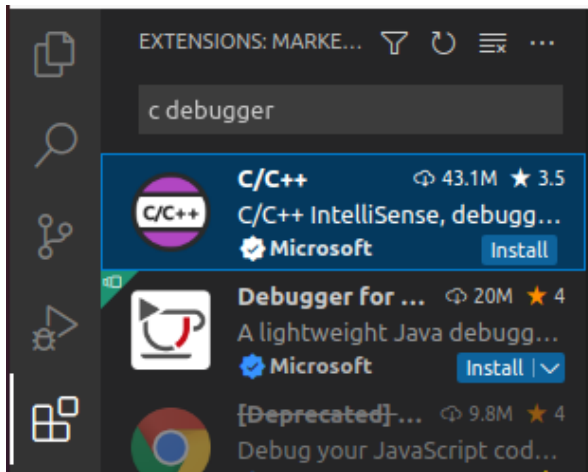
Kompajliranje i pokretanje programa

1. U Terminalu prethodno otvorenog VS Code editora (na putanji `pjisp`)
2. Kreirati novi direktorijum pomoću komande `mkdir v1`, ući u njega `cd v1`
3. Kreiranje fajla pomoću komande `code <ime_fajla>.c`
 - Obavezno sa ekstenzijom `.c`
4. Pisanje (izmena) koda u kreiranom fajlu
5. Čuvanje sadržaja fajla
6. Kompajliranje fajla pomoću komande `gcc <ime_fajla>.c -lm`
 - Ukoliko kompajler ispiše tekst na komandnoj liniji, znači da je došlo do greške pri kompajliranju. U tom slučaju, vratiti se na 3. korak
7. Pokrenuti dobijeni program sa sledećom komandom: `./a.out`

Podešavanje dibagera

- Dibager (debugger) - program koji omogućava izvršenje programa korak po korak i praćenje njegovih vrednosti

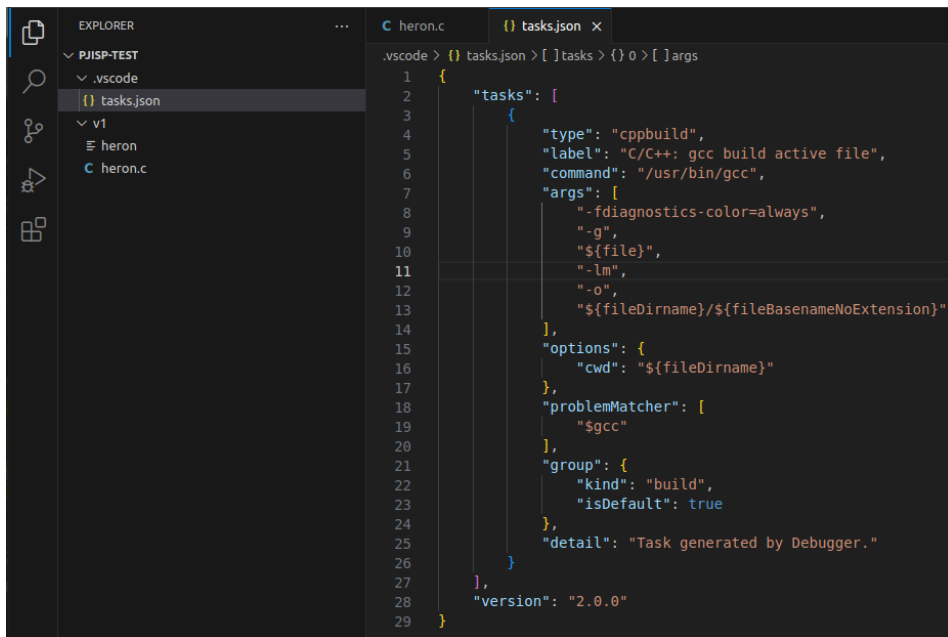
Dodati podršku u VS Code instaliranjem ekstenzije:



Pokretanje dibagera

- Otvoriti u VS Code editoru `heron.c` i kliknuti sa leve strane broja linije tako da se pojavi crvena tačkica (breakpoint)
- Izabrati opciju Run -> Start Debugging (prilikom prvog pokretanja izabrati verziju gcc-a bez sufiksa)
- Prethodna ekstenzija radi bez problema ukoliko nema `math.h` kao zavisnosti u programu
 - Da bi bili podržani i programi koji koriste `math.h`, uraditi File -> Open Folder i izabrati prethodno kreirani `pjisp` direktorijum gde će se nalaziti sve vežbe
 - U Explorer-u će se pojaviti `pjisp` direktorijum i unutar njega `.vscode` sa datotekom `tasks.json` koji je potrebno modifikovati
 - Dodati opciju `"-lm"` pre opcije `"-o"` (11. linija) i sačuvati izmenu
 - Ako je sve urađeno kako treba, biće moguće debugovati `heron.c` (isprobati opcije Continue, Step Over i Stop)

Primer izgleda tasks.json datoteke



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure: a workspace named 'PJISP-TEST' containing a '.vscode' folder with 'tasks.json', and a 'v1' folder containing 'heron' and 'heron.c'. The 'tasks.json' file is selected and its content is displayed in the main editor. The editor shows the following JSON configuration:

```
.vscode > {} tasks.json > [ ] tasks > { } 0 > [ ] args
1  {
2      "tasks": [
3          {
4              "type": "cppbuild",
5              "label": "C/C++: gcc build active file",
6              "command": "/usr/bin/gcc",
7              "args": [
8                  "-fdiagnostics-color=always",
9                  "-g",
10                 "${file}",
11                 "-lm",
12                 "-o",
13                 "${fileDirname}/${fileBasenameNoExtension}"
14             ],
15             "options": {
16                 "cwd": "${fileDirname}"
17             },
18             "problemMatcher": [
19                 "$gcc"
20             ],
21             "group": {
22                 "kind": "build",
23                 "isDefault": true
24             },
25             "detail": "Task generated by Debugger."
26         }
27     ],
28     "version": "2.0.0"
29 }
```

Primeri deklaracije promenljivih

- `int i;`
- `char c;`
- `float f1, f2;`
- Sa definisanim početnim vrednostima
 - `float f1 = 7.0, f2 = 5.2;`
 - `unsigned int ui = 0;`

Tipovi podataka u programskom jeziku C

- `char` - znakovni tip, dužine 1 bajta
- `int` - celobrojni tip, dužine 4 bajta
 - `short int` (ili samo `short`) - 2 bajta (ređe korišćen)
 - `long int` (ili samo `long`) - 4 ili 8 bajtova (ređe korišćen)
 - `signed` i `unsigned`
- `float` - realni tip jednostruke preciznosti, dužine 4 bajta
- `double` - realni tip dvostruke preciznosti, dužine 8 bajtova
 - `long double` - 8 bajtova (ređe korišćen)

Funkcije za učitavanje i ispis

- `printf` - ispis na standardni izlaz (ekran)
- `scanf` - učitavanje sa standardnog ulaza (tastatura)
- Omogućavaju formatiranje ulaznih i izlaznih podataka

Primer 1

- Učitavanje temperature u Celzijusima i konverzija u Kelvine

```
#include <stdio.h>

int main()
{
    double celzijusi, kelvini;

    printf("Unesite temperaturu u Celzijusima: ");
    scanf("%lf", &celzijusi);
    kelvini = celzijusi + 273.15;
    printf("%lf stepeni Celzijusa je %lf stepeni Kelvina\n",
           celzijusi, kelvini);

    return 0;
}
```

Format specifikatori

- Konstrukcija %<format> se u printf i scanf funkciji zamenjuje vrednošću odgovarajuće promenljive
 - %c - karakter konverzija (char)
 - %d - celobrojna konverzija (int)
 - %u - neoznačena celobrojna konverzija (unsigned)
 - %f - konverzija realne vrednosti (float)
 - %lf - konverzija realne vrednosti dvostruke preciznosti (double)
 - %g - drugačiji format prikaza realne vrednosti dvostruke preciznosti
 - %% - ispisivanje karaktera koji označava procenat

Zadatak 2

Proširiti Zadatak 1 tako da se vrednosti stranica trougla unose sa tastature. Prikaz rezultata površine trougla zaokružiti na dve decimale.

Zadatak 3

Napisati program kojim se vrši prevođenje količine tečnosti iz galona u litre, ako je 1 galon 4.54 litra. Količina tečnosti u galonima se unosi sa tastature.

Zadatak 4

Jedan radnik određeni posao uradi za M dana, a drugi radnik isti posao uradi za N dana. Napisati program kojim se određuje za koliko dana bi taj posao bio završen ako bi radili zajedno. Vrednosti M i N se unose sa tastature.

Zadatak 5

Napisati program koji prihvata broj dana kao celobrojnu vrednost i na osnovu nje izračunava ukupan broj godina, meseci i dana. Uzeti da godina ima isključivo 365, a mesec 30 dana.

Dodatni zadaci za vežbu

- Zbirka: "Rešeni zadaci iz programskog jezika C", Laslo Kraus
- Zadaci:
 - 1.1 Ispisivanje teksta na glavnom izlazu
 - 1.2 Izračunavanje zbira dva cela broja
 - 1.3 Računanje obima i površine kruga
 - 1.4 Izračunavanje površine trougla u ravni