1. Nabrojte sve promenljive, tipove podataka i petlje koje smo koristili dosad i ukratko opišite svaku pomenutu stavku ponaosob.

**Vrste promenljivih i tipovi podataka:**

**byte** - pise se malim slovima, sluzi za dodeljivanje vrednosti celih brojeva od -128 do 127, zauzima minimalno memorijskog prostora, ukoliko se pojavi ostatak (decimalan zapis) taj deo se odseca

**short** - pise se malim slovima, sluzi za dodeljivanje vrednosti celih brojeva od -32,768 do 32,767, zauzima vrlo malo memorijskog prostora, ukoliko se pojavi ostatak (decimalan zapis) taj deo se odseca

**int** - pise se malim slovima, sluzi za dodeljivanje vrednosti celih brojeva od -2,147,483,648 do 2,147,483,647, ne zauzima mnogo memorijskog prostora, ukoliko se pojavi ostatak (decimalan zapis) taj deo se odseca

**long** - pise se malim slovima, sluzi za dodeljivanje vrednosti velikih celih brojeva od -9,223,372,036,854,775,808 do 9,223,372,036,854,775,807, zauzima vise memorijskog prostora od svih gore navedenih

**float** - pise se malim slovima, sluzi za dodeljivanje vrednosti decimalnih brojeva, zahteva vise memorijskog prostora od int-a

**double** - pise se malim slovima, sluzi za dodeljivanje vrednosti decimalnih brojeva, zahteva vise memorijskog prostora od float-a jer moze da primi vise decimala (duplo je precizniji, pa otud i naziv double)

**char** - pise se malim slovima, dodeljivanje samo jednog karaktera i njegova definicija se pise izmedju pojedinacnih navodnika ‘ ‘

**String** - pise se pocetnim velikim slovom, sluzi za dodeljivanje niza karaktera (teksta) i njegova definicija se pise izmedju duplih navodnika “ “

**Boolean** - pise se malim slovima, sluzi za definisanje 2 stanja - true ili false (da li je neka tvrdnja tacna ili nije), zauzima najmanje memorijskog prostora od svih navedenih (1bit).

**Petlje:**

**if** - nakon izraza *if* u malim zagradama se postavlja uslov za ulaz u petlju. Ukoliko se uslov ispunjava, ulazi se u deo koda koji je u viticastim zagradama, a ukoliko uslov nije ispunjen, preskace se taj kod.

**if … else** - nakon izraza *if* u malim zagradama se postavlja uslov za ulaz u petlju. Ukoliko se uslov ispunjava, ulazi se u deo koda koji je u viticastim zagradama, a ukoliko uslov nije ispunjen, izvrsava se deo koda koji se nalazi unutar viticastih zagrada nakon komande *else*.

**if … else if … else** - nakon izraza *if* u malim zagradama se postavlja uslov za ulaz u petlju. Ukoliko se uslov ispunjava, ulazi se u deo koda koji je u viticastim zagradama, a ukoliko uslov nije ispunjen, proverava se novi uslov iz malih zagrada nakon *else if* komande. Ako je ovaj uslov ispunjen izvrsava se kod iz viticastih zagrada. Ukoliko ni taj uslov nije ispunjen, izvrsava se deo koda koji se nalazi unutar viticastih zagrada nakon komande *else*.

**switch** - nakon izraza *switch* u malim zagradama se postavlja uslov za ulaz u petlju. U njoj se nalazi vise *case*-ova (opcija). Uslov se poredi sa svakim *case*-om i ako se poklapa, izvrsava se kod iz tog *case*-a. Ukoliko se uslov ne slaze ni sa jednim *case*-om, izvrsava se deo koda iz dela *default* (ukoliko je definisan). Svaki *case* sadrzi (ali ne mora) komandu *break* kako bi se prekinulo dalje izvrsavanje ostalih *case*-ova.

**while** - nakon izraza *while* u malim zagradama se postavlja uslov za trajanje petlje. Dokle god je uslov true, petlja se izvrsava.