

Основи алгоритмизације

Алгоритам је поступак долажења до решења неког проблема. Састоји се од коначног низа корака који се одвијају одређеним редоследом.

Алгоритам нам може помоћи не само у случају решавања проблема помоћу рачунара, већ и у решавању многобројних једноставних активности које су уобичајене у нашој свакодневици, као што је случај са припремањем кулинарских специјалитета, чаја и слично.

Алгоритам конструишемо тако што почињемо од једноставне, уопштене идеје, а потом поједине кораке поступно разлажемо. То значи да главни проблем рашчлањујемо на мање проблеме које треба решавати.

Основни кораци алгоритма.



Слика 1.4. Дијаграм основних корака алгоритма

Описивање алгоритма.

Постоји више начина да се опише алгоритам: природним језиком, псеудо језиком, дијаграмом и [програмом](#). У наставку ћемо објаснити сваки од ових типова.

ТИП А. Опис природним језиком

Описати алгоритам природним језиком значи детаљно, јасно, прецизно, недвосмислено и тачним редоследом описати кораке који воде решењу проблема.

Пример 1. Опишимо природним језиком алгоритам за сабирање произвољна два броја. Нека то буду a и b за бројеве које сабирамо и c за резултат.

1. Унети вредности за променљиве *a* и *b*.
2. Израчунати збир променљивих *a* и *b* и доделити их променљивој *c*.
3. Приказати вредност променљиве *c*.

Пример 2. Приметимо да и кување кафе такође можемо представити алгоритмом.

1. Узети лонче и сипати воду и шећер по потреби.
2. Укључити ринглу.
3. Сачекати да вода прокључа.
4. Кад вода прокључа, сипати кафу и промешати.
5. Сачекати да се ниво напитка подигне и тада склонити лонче са рингле.
6. Искључити ринглу.
7. Сипати кафу у шољицу.



ТИП Б. Опис псеудо језиком

Псеудо језик представља комбинацију природног језика и неког програмског језика. Код овог типа описивања алгоритма веома је важно користити исте језичке конструкције на исти начин. Такође, уколико је потребно, псеудокод треба да буде праћен одговарајућим објашњењима.

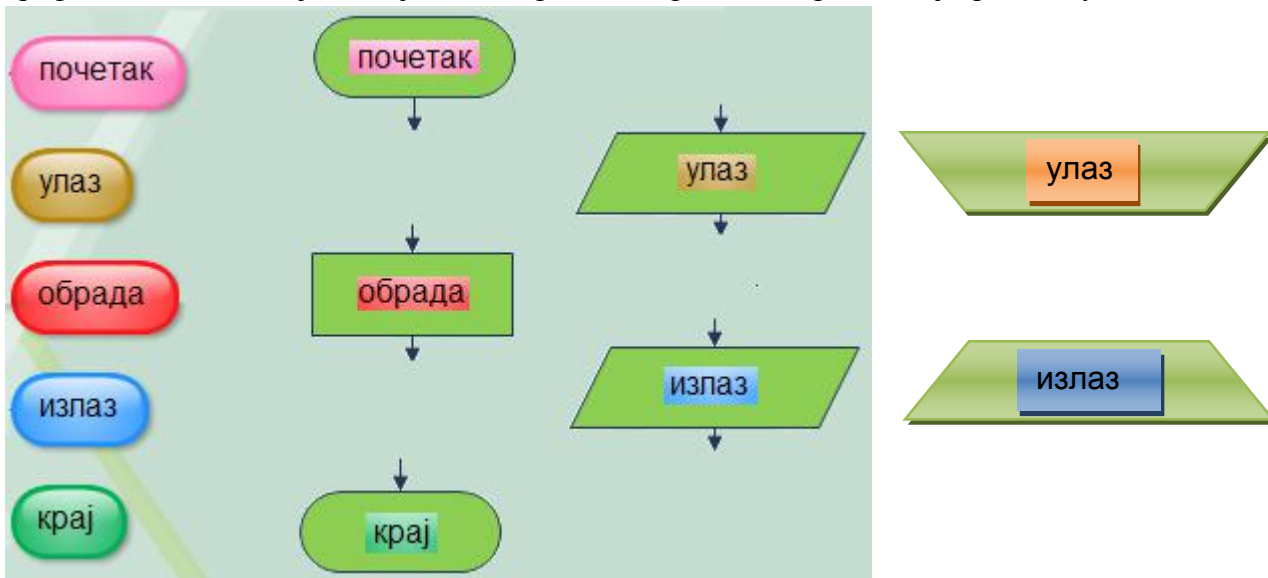
Пример 3. Опишимо псеудо језиком алгоритам за сабирање произвољна два броја. Нека то буду *a* и *b* за бројеве које сабирамо и *c* за резултат, као код *примера 1*.

```
Почетак  
Упиши a и b  
c = a + b  
Испиши c  
Крај
```

ТИП В. Опис алгоритма блок шемом (дијаграм)

Код овог облика описивања алгоритма користе се међународно договорени симболи прописани ISO стандардом, а ток рада алгоритма представља се линијама које повезују симболе. Овај тип описивања има предност у односу на описивање псеудо језиком и природним језиком јер не зависи од говорног језика онога ко саставља алгоритам, графички приказ је једноставан, прегледан и лако је пронаћи грешке.

Графички симболи који се најчешће користе за приказ алгоритма дијаграмом су:



Слика 1.5. Графички приказ неких основних корака алгоритма

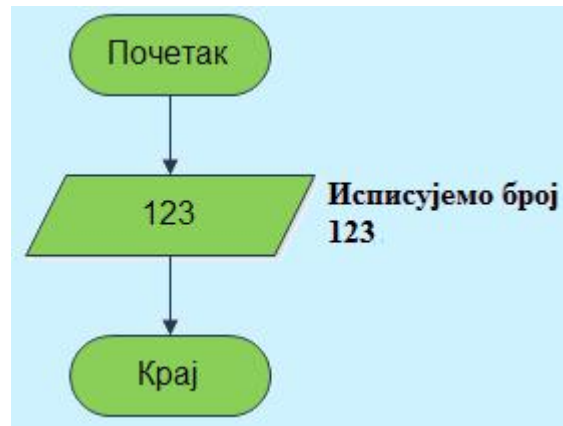
Пример 4. Покушајмо да опишемо дијаграмом алгоритам који испишује поруку „Здраво“ на екрану.



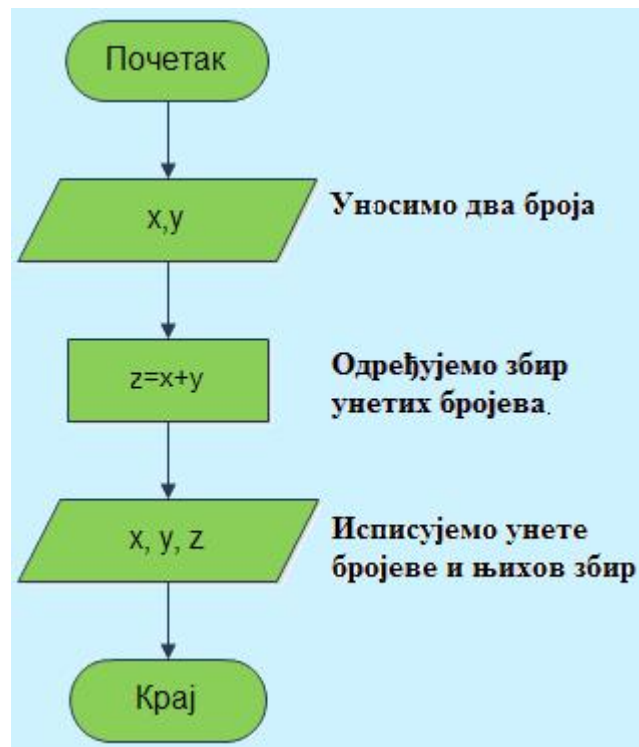
Пример 5. Покушајте да сами опишете дијаграмом алгоритам за исписивање имена и презимена, на пример Пера Перић.



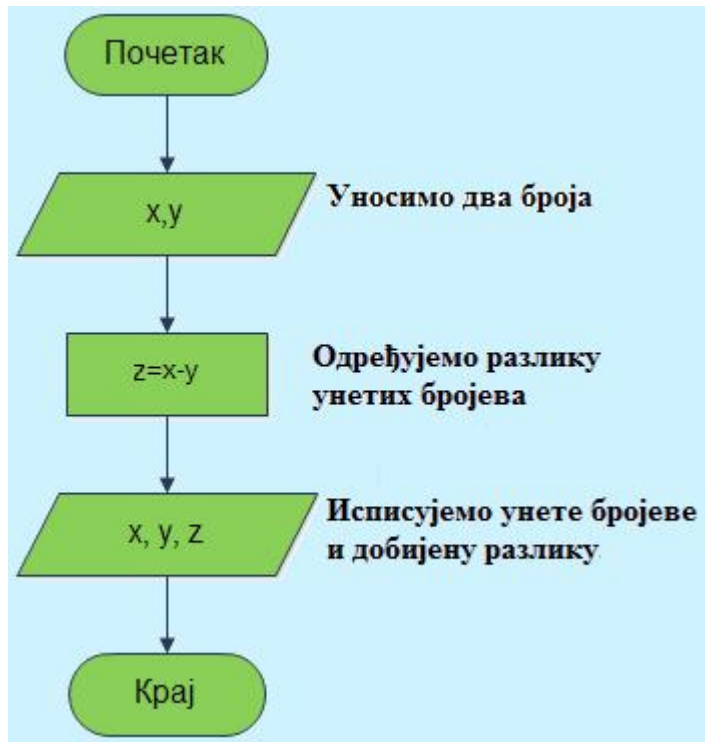
Пример 6. Покушајте да опишете дијаграмом алгоритам за исписивање броја 123.



Пример 7. Опишимо дијаграмом алгоритам за сабирање два произвољна броја.



Пример 8. Опишите дијаграмом алгоритам за разлику, производ и количник два произвољна броја, а потом и квадрат задатог произвољног броја.



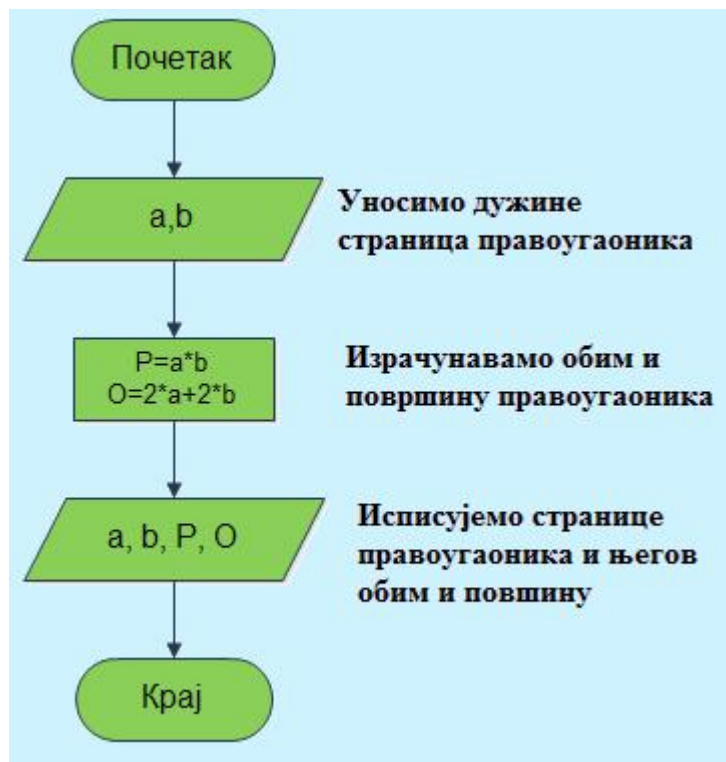
Пример 9. Опишите алгоритам (дијаграмом) за израчунавање израза $x = a / 3$, a је дат произвољан број.



Пример 10. За дати полупречник напишите алгоритам за израчунавање површине круга.

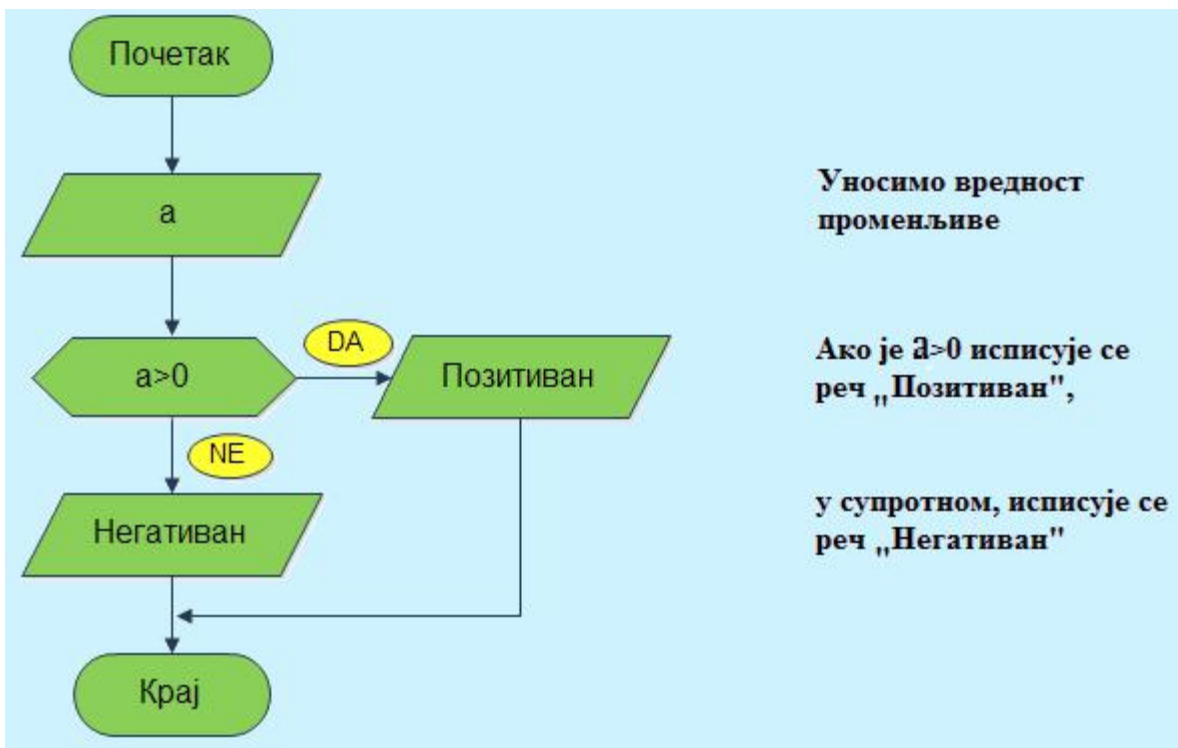


Пример 11. Дате су дужине страница правоугаоника a и b . Дијаграмом опишите алгоритам за рачунање површине и обима правоугаоника датих страница, а онда испишите дужину страница, површину и обим правоугаоника.

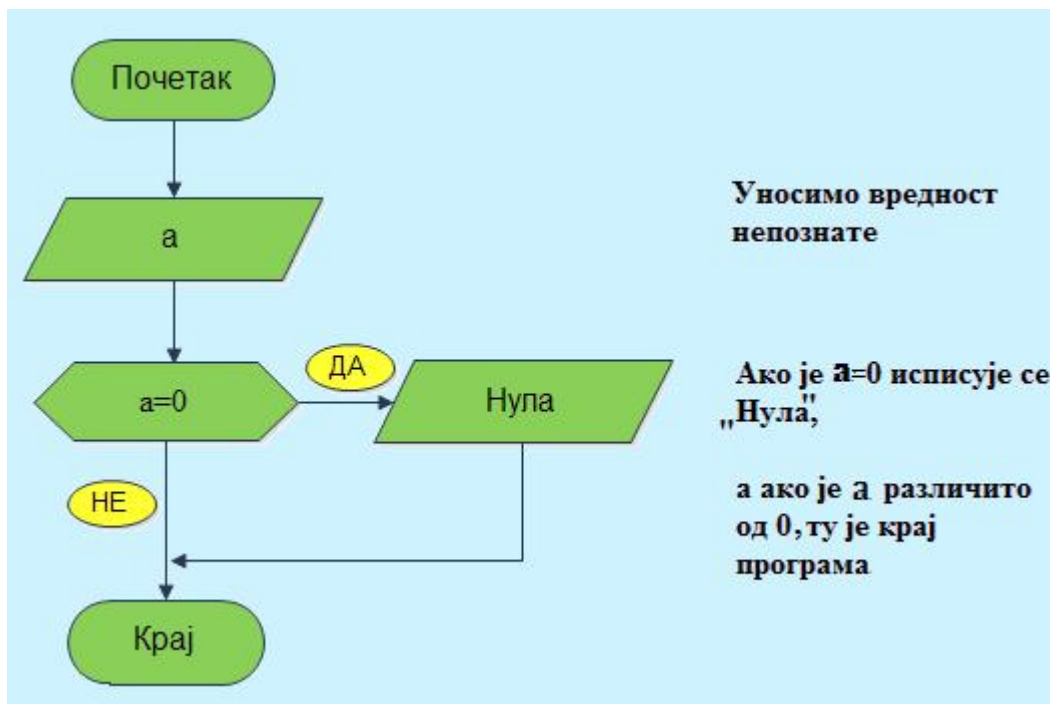


Шта се дешава уколико се од нас тражи да напишемо алгоритам који извршава одређени задатак, али само уколико испуњава одређени услов? Погледајмо следећи пример.

Пример 12. Дат је произвољан број различит од нуле. Напишимо алгоритам који испишује да ли је број позитиван или негативан.



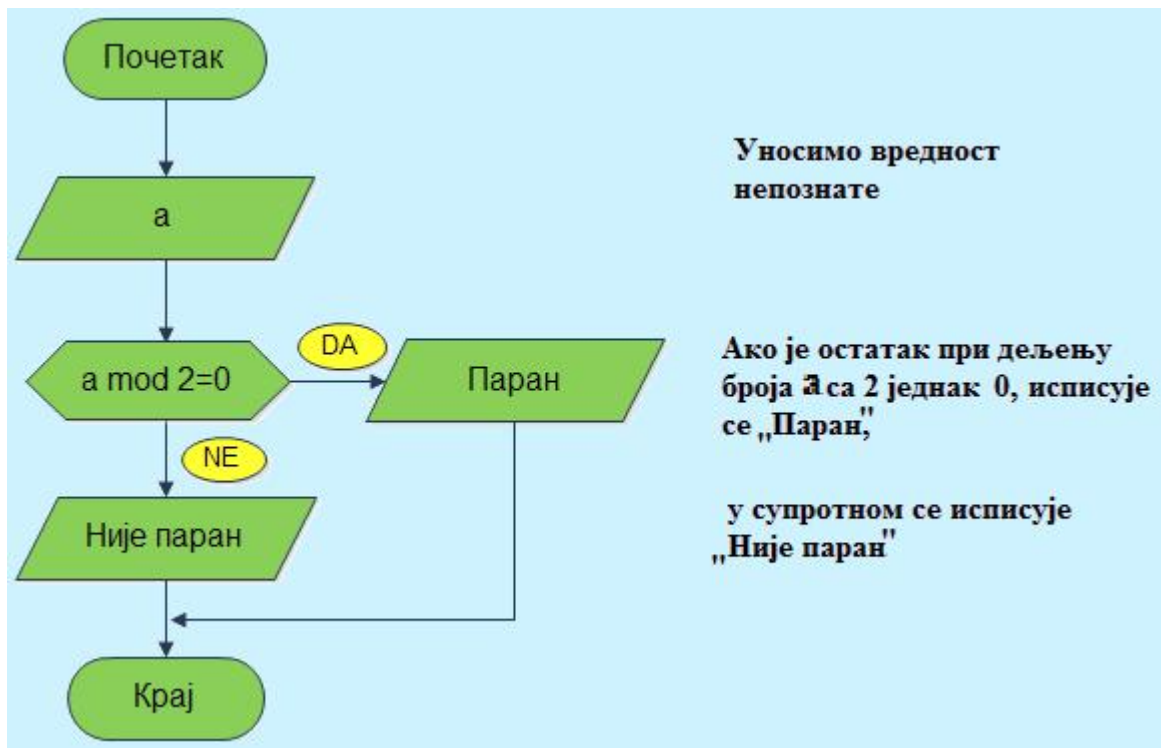
Пример 13. Дат је произвољан број. Напишите алгоритам који проверава да ли је број једнак нули и испишује „Нула” ако јесте.



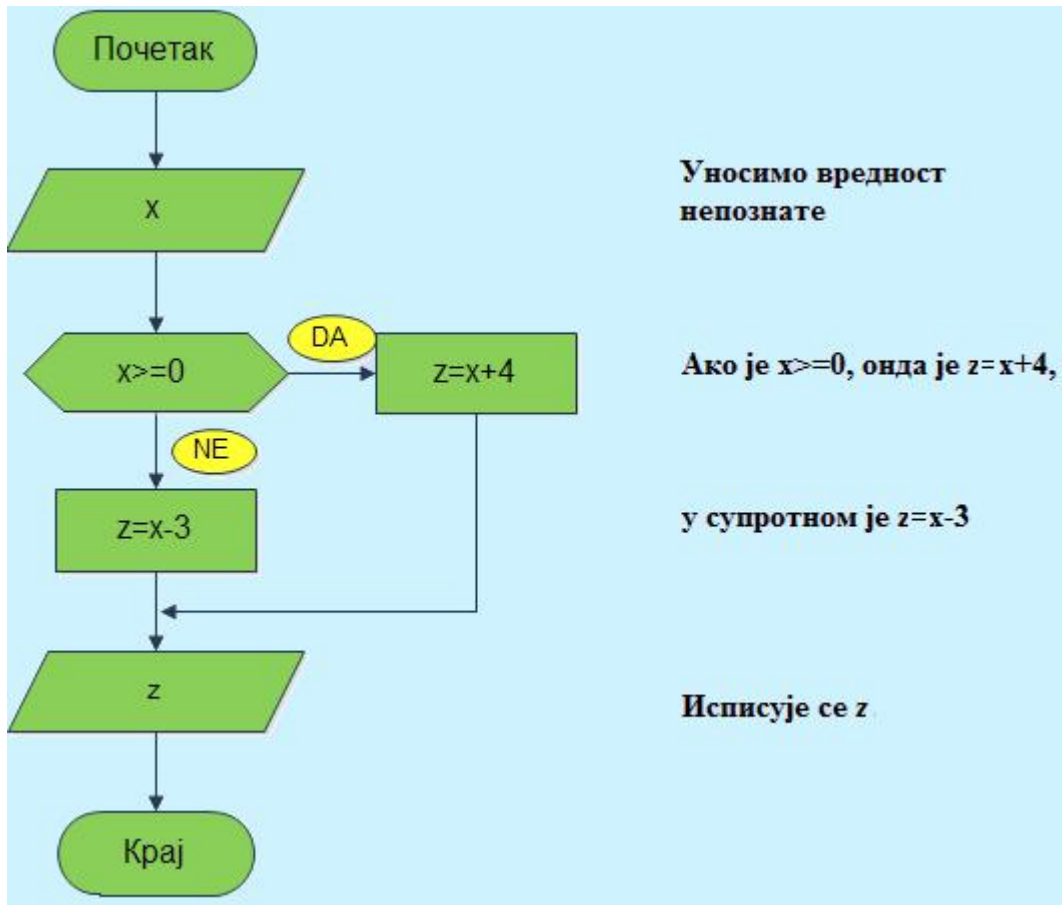
Пример 14. Напишите алгоритам који проверава да ли је од два дата броја први дељив другим, а потом исписује „Дељив“ ако јесте.



Пример 15. Напишите алгоритам који проверава да ли је дати произвољни број паран или није, а потом то и исписује.

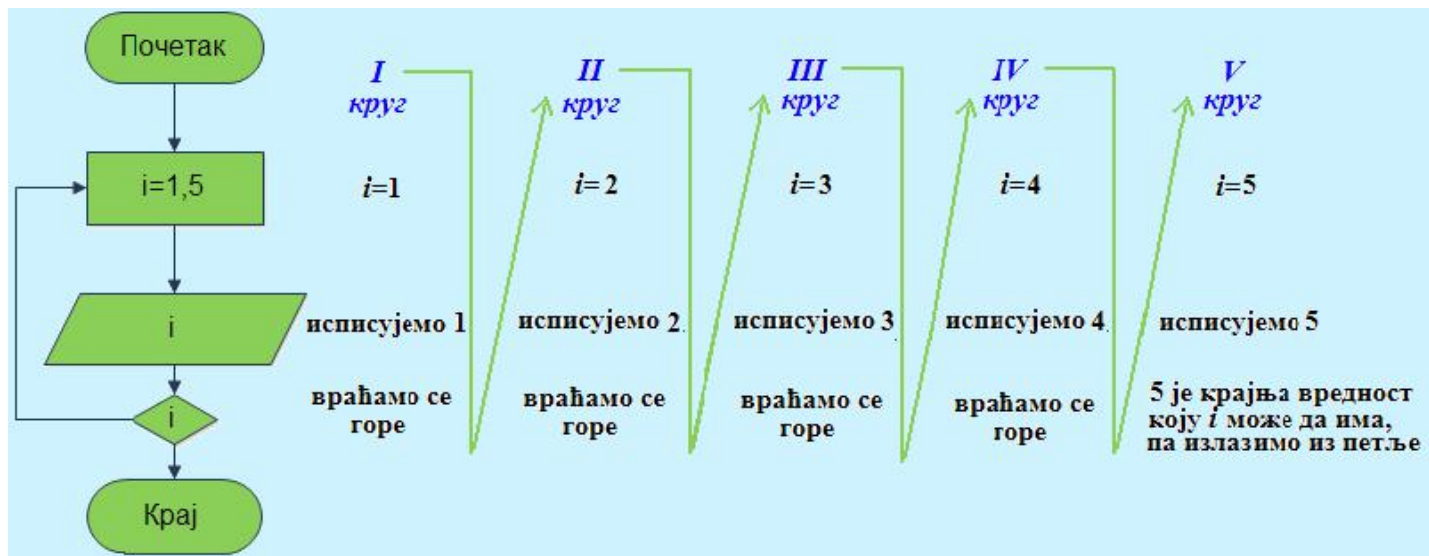


Пример 16. Дат је произвољан број **x**. Напишите алгоритам који исписује број за 3 мањи од **x**, ако је **x** негативан број, а у противном исписује број који је за 4 већи од **x**.

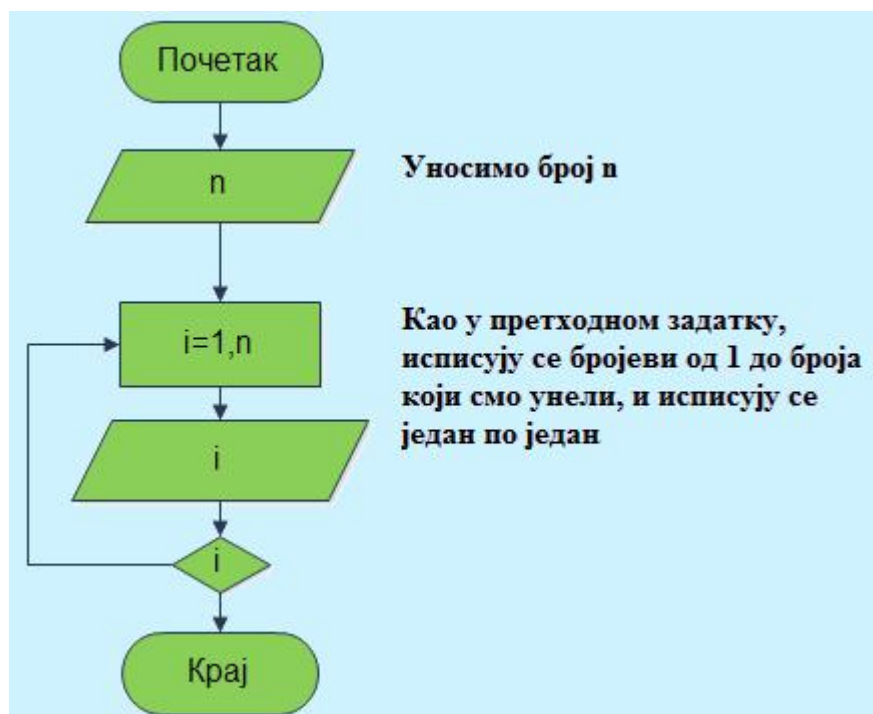


Често се дешава да нам се тражи у неком задатку да више пута поновимо исту радњу. Код програмирања и алгоритмизације се то на једноставан начин може избећи. У следећем примеру се од нас тражи да напишемо алгоритам који ће исписивати све природне бројеве од 1 до 5. Ово би могло да се реши тако што ћемо уносити редом све те бројеве, слично као у примеру 6, али проблем настаје када се од нас тражи да унесемо и испишемо првих 1000 бројева, на пример. Погледајмо како се такви проблеми једноставно решавају.

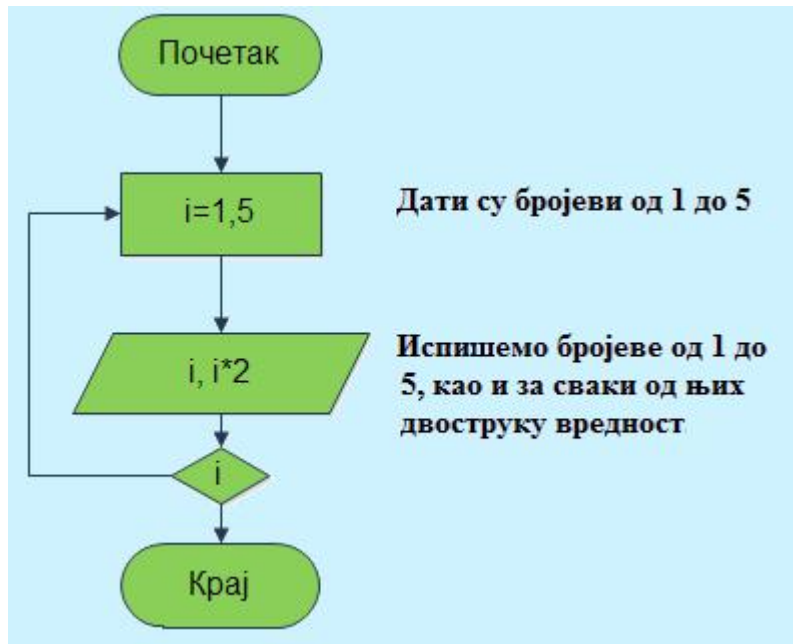
Пример 17. Написати алгоритам који исписује редом све природне бројеве од 1 до 5.



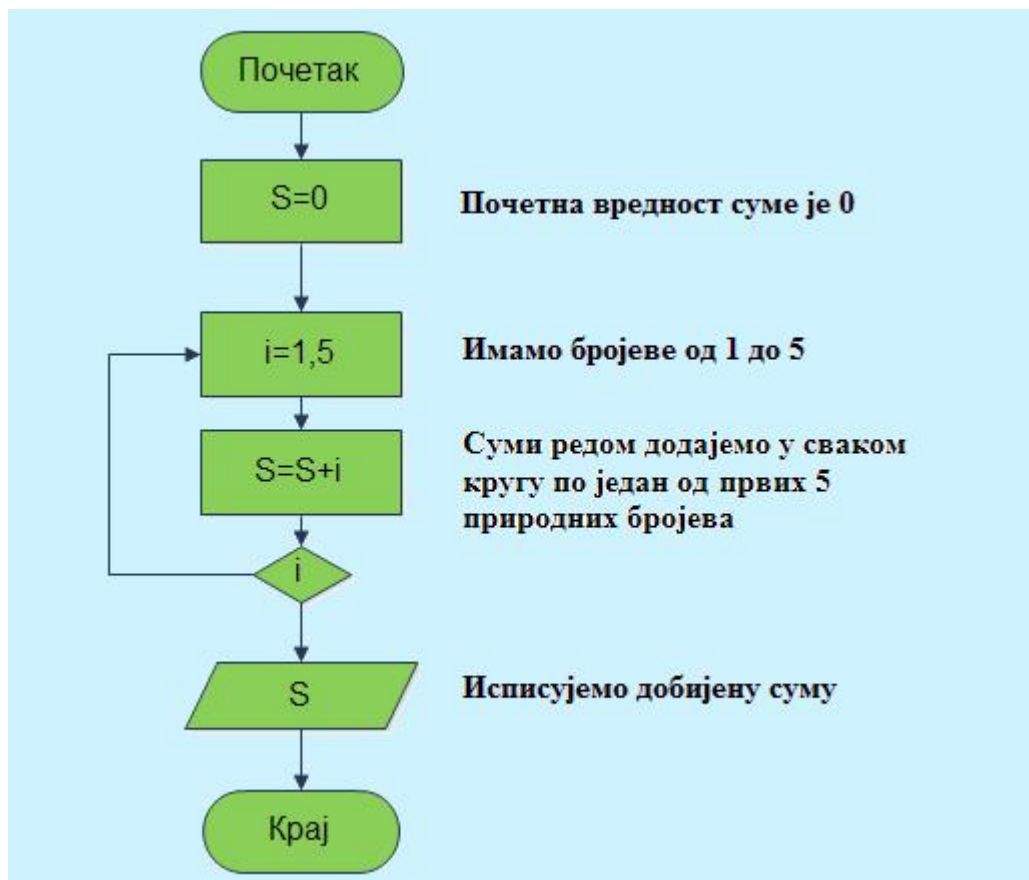
Пример 18. Напишите алгоритам који за произвољно n које се задаје испишује првих n природних бројева.



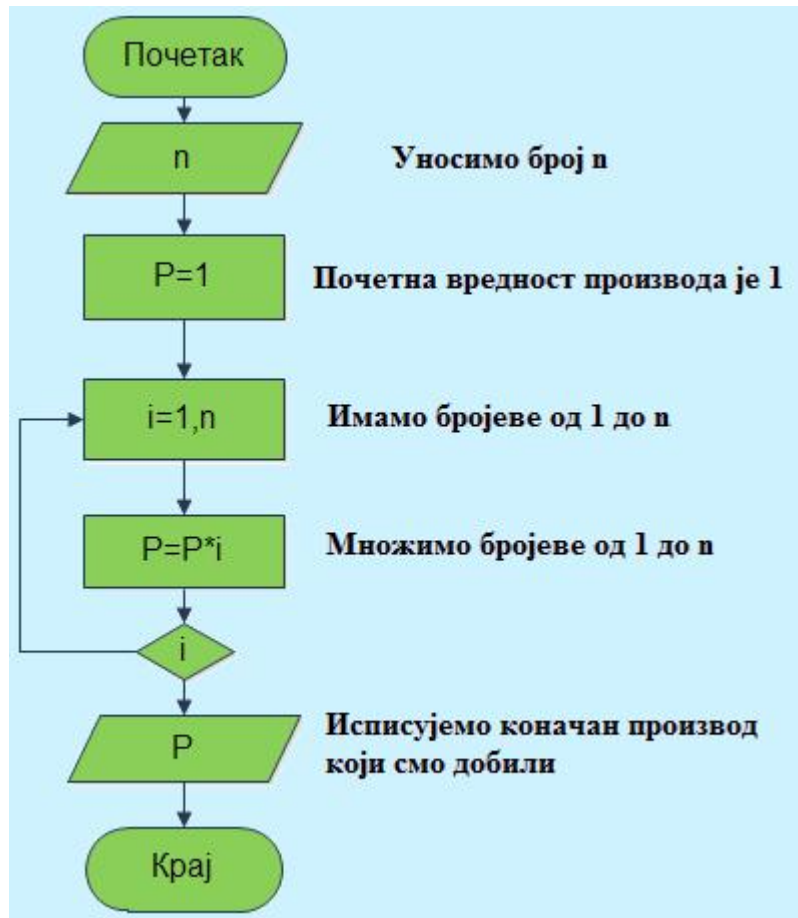
Пример 19. Напишите алгоритам који испишује редом првих пет природних бројева заједно са њиховом двоструком вредношћу.



Пример 20. Опишите дијаграмом алгоритам који исписује суму првих пет природних бројева.



Пример 21. Описите дијаграмом алгоритам који исписује производ првих n природних бројева. Број n је задат и произвољан природни број.



Пример 22. Дата су два природна броја, мањи k и већи n . Напишите алгоритам за исписивање аритметичке средине бројева од k до n укључујући и те бројеве.

