1. Svaki algoritam zahteva odredjene tipovi podataka. Ponekad treba koristiti razlomacki brojevi, onda naprimjer u python koristimo float. Ili recimo rad sa nekim tesktom. Ako ne bih bio stringova, kako bih smo cuvali ti teskovi? Kroz nizovi i cjeli brojevi koji bi predstavljali vrednosti iz ASCII tabele itd. Ali koliko vremena to zauzmi i je li to udobno? Zbog razlitih ciljeva koristimo razliciti tipovi. Ovo je ubrzava razradu algoritama i opste programa. I svaki tip ima odredjenu velicinu, za int 32 bita, float 64 bita, boolean 8 bit.
2. Postoji. Ovo je nesto, sto moze generisati kod visih programskih jezika ili naprimjer neki jos vise nivo, kao Python ili Java iznad C++ i C. O posljednjem još ne znam, moguće je da postoje neki konstruktori kao što je BAS (Browser Automation Studio) iznad JS (JavaScript), ali generistai kod moze ChatGPT i ostale AI/LLM. U rijeci apstrakcija postoji nekiloko znacanja. U smislu programskih jezika to znaci neki nivo koji se nalazi iznad drugog. Prehod na novi nivo olaksava razradu zbog sakrivanja odredjenih operacija kao sto je alokacija memorija, ali kao pravilo smanjuje brzinu.
3. Varijabla temelj programiranja. Uz njenu pomoc imamo priliku pristupiti ka podacima koji se nalaze u memoriji i izvrsiti odredjene manipulacije sa nima.
4. Immutable. Znaci nakon promjene variabli se kreira novi sadrzaj, ova promjena povezana sa lokacijom podataka u novu memorijsku ćeliju. Sa drugoj strani, naprimjer za integer mjenja samo vrijednost, adresa u memoriji se ostaje ista. U vecini programskih jezika stringovi su nepromjenlivi tipovi podataka zbog bezbjednosti prenosa podataka naprimjer kroz fajlovi ili niti. Odnosno ima zastita od promjene, znaci manja vjerovatnoca da izgubimo ti podatke. Pored toga immutable daje priliku za keshiranje sto oznacava da mozemo cuvati isti podatke na jednoj memorijskoj lokaciji.
5. A) TRUE B) FALSE C) TRUE D) FALSE E) FALSE
   1. s = ‘popravni ‘ + s
   2. res = s[-8:]
   3. as je vanredni is
   4. s[::-2]
   5. s[-3:]
   6. s[len(s) // 2 - 1 : len(s) // 2 + 1]
   7. s[1] + s[len(s) // 2 - 1 : len(s) // 2 + 1] + s[-2]
   8. Za cuvanje skupa podataka. Ovo nam daje lako upravljanje kroz jednu varijablu. Ako pretpostavimo da nemamo liste, da li bi bilo udobno inicializovati hiljadu varijabli i nakon toga upravljati hiljadom varijabli?
   9. 3
   10. 3
   11. 3
   12. Varijabla – povezuje ime sa adresoj u memoriji. Cuva podatke I omogucava manipulaciju.
   13. Algoritam – skup odrdjenih koraka koji dovode do jasnog rezultata.
   14. Interpretator – program koji izvrsava kod po jednoj naredbe. U pajtonu prvo prevodi naredbu u bajt kod I zatim virtuelna masina prevodi taj bajt kod u masinski.
   15. Compajler – program koji prevodi ceo program u bajt kod, nakon toga ovaj fajl sa bajt kodom naprimjer u javi izvrsava virtuelna masina.
   16. Int – tip podataka koji cuva cijeli brojevi od -2^32 do 2^32 -1.
   17. Lista – struktura podataka koja cuva skup vrijdenosti u jednoj varijabli
   18. Git – alat za upravljanje kontrolom verzija
   19. GitHub – platforma na kojoj se nalaze programe (repositories)
   20. Funkcija – sadrzi kod koji mozemo pozivati preko nje koliko god puta u programu
   21. Petlja – konstrukcija koja se koriste kada treba ponoviti izvrsavanje bloka koda.