1. Napišite program koji traži od korisnika da unese listu celih brojeva. Program treba da uradi sledeće:

* Štampa ukupan broj podataka u listi.
* Štampa poslednji podatak u listi.
* Štampa listu obrnutim redosledom.
* Štampa Da ako lista sadrži 5 na Ne ako ne sadrži.
* Štampa broj petica u listi.
* Briše prvi i poslednji podatak iz liste, sortira preostale podatke i štampa rezultat.
* Štampa koliko celih brojeva u listi je manje od 5.

1. Napišite program koji generiše listu od 20 slučajnih brojeva između 1 i 100, pa:

* Štampa listu.
* Štampa srednju vrednost svih elemenata iz liste.
* Štampa najveću i najmanju vrednost iz liste.
* Štampa drugu najmanju i drugu najveću vrednost iz liste.
* Štampa koliko ima parnih brojeva u listi.

1. Počnite sa listom [8,9,10]. Uradite sledeće:

* Postavite drugi element (indeks 1) na 17
* Dodajte 4, 5, i 6 na kraj liste
* Brišite prvi element liste
* Sortirajte listu
* Duplirajte listu
* Ubacite 25 na indeksu 3

Na kraju treba da dobijete listu [4,5,6,25,10,17,4,5,6,10,17]

1. Tražite od korisnika da unese listu koja sadrži brojeve između 1 i 12. Zatim zamenite sve brojeve koji su veći od 10 sa brojem 10.
2. Tražite od korisnika da unese listu stringova. Kreirajte novu listu koja sadrži iste stringove ali bez prvog slova.
3. Kreirajte sledeće liste korišćenjem petlji.

* Listu koja se sastoji od brojeva 0 do 49
* Listu koja sadrži kvadrate brojeva od 1 do 50.
* Listu [‘a’, ‘bb’, ‘ccc’, ‘dddd’, … ] koja se završava sa 26 kopija slova z.

1. Napišite program koji uzima dve liste L i M, iste dužine, dodaje njihove elemente da kreira novu listu N čiji su elementi sume odgovarajućih elemenata iz L i M. Na primer, ako je L=[3,1,4] i M=[1,5,9], onda će N biti [4,6,13].
2. Napišite program koji traži od korisnika da unese ceo broj, pa kreira listu koja se sastoji od faktora tog broja.
3. Kada igrate igru u kojoj se koriste dve kockice, valja znati kolike su šanse za pojavljivanje nekog ishoda. Na primer verovatnoća za dobijanje 12 je oko 3%, a verovatnoća za dobijanje 7 je oko 17%. Mogli bi izračunati ovo matematički, ali ne znate verovatnoću. Međutim, znate na napišete program kojim ćete simulirati bacanje dve kockice 10000 puta i izračunati koliki je procenat pojavljivanja ishoda 2, 3, 4, …, 12.
4. Napišite program koji rotira elemente liste tako da se element na prvom indeksu pomera na drugi, drugi na treći, itd., sve do poslednjeg koji se pomera na prvi indeks.
5. Korišćenjem for petlje kreirajte listu prikazanu dole, koja se sastoji od jedinica razdvojenih rastućim brojem nula. Poslednje dve jedinice treba da budu razdvojene sa deset nula.

[1,1,0,1,0,0,1,0,0,0,1,0,0,0,0,1,**....**]

12. Napišite program koji generiše 100 slučajnih brojeva koji su ili 0 ili 1. Zatim odredite najduži uzastopni niz nula. Na primer, najduži uzastopni niz nula u listi [1,0,1,1,0,0,0,0,1,0,0] je 4.

* 1. Napišite program koji briše višestruka pojavljivanja podataka u listi, tako da se podaci pojavljuju samo jednom. Na primer lista [1,1,2,3,4,3,0,0] će posle takvog brisanja postati [1,2,3,4,0].
  2. Napišite program koji traži od korisnika da unese dužinu u feet-ima. Program treba da korisniku pruži mogućnost da bira jedincu u koju će ta dužina biti konvertovana. Jedinice mogu biti inči, jarde, milje, milimetri, centimetri, metri ili kilometri. Iako ovo može biti urađeno if naredbama, mnogo je kraće ako koristimo liste i for petlje, a takođe je mnogo lakše ako želimo da dodamo nove jedinice za konverziju.
  3. Postoji neprobojna šifra pod nazivom „jednokratni ključ“. Način na koji ona radi je da pomerate svako slovo poruke za slučajan broj između 1 i 26, sa preklapanjem azbuke ako je potrebno. Na primer ako je slovo koje treba šifrirati y a slučajni pomak 5, tada je novo slovo d. Svako slovo dobija svoj pomak, pa nam je potrebno onoliko slučajnih pomaka koliko ima slova u poruci. Kao primer recimo da je korisnik uneo reč secret . Program treba da generiše 6 slučajnih brojeva između 1 i 26. Pretpostavimo da su generisani brojevi 1, 3, 2, 10, 8, i 2. Šifrovana poruka će biti thebmv.
  4. Napišite program koji traži od korisnika da unese poruku, pa tu poruku šifrira korišćenjem gornjeg metoda. Najpre konvertujte string u mala slova. Svaki blanko znak i znak interpunkcije treba da ostane nepromenjen. Na primer Secret!!! postaje thebmv!!! korišćenjem gornjih pomaka.
  5. Napišite program koji dešifruje poruke šifrovane na gornji načina.

Razlog zašto se ovaj metod zove „jednokratni ključ“ je zato što se slučajni pomaci koriste samo jedanput. Šifra bi mogla biti lako razbijena ako bi se isti pomaci koristili u više poruka. Čak štaviše, ovaj metod je potpuno siguran samo ako su slučajni brojevi stvarno slučajni što nije slučaj sa randint funkcijom koja daje takozvane pseudo slučajne brojeve. Za potrebe kriptografije postoje drugi pouzdaniji načini generisanja slučajnih brojeva.