***Naloge iz kazalcev na polja in strukturna polja (tabele)***

1. Deklariraj niz beseda, ki naj vsebuje največ 20 znakov. Vrednost niza vnesi s tipkovnice ter s pomočjo kazalca na niz omogoči dodajanje znakov (brez uporabe že napisanih metod ali funkcij) v niz beseda, na n-to mesto.Primer:

* beseda: Puto , vstavljanje na 3. mesto, črka 'n': Punto

1. Deklariraj kazalec na niz *ime* in kazalec na niz *priimek.* Oba niza beri s tipkovnice ter ju s pomočjo omenjenih kazalcev (brez uporabe že napisanih metod ali operatorjev) združi in izpiši na naslednji način:

* vnos: Janez(ime) Novak(priimek) , združeni niz: J. Novak

1. V programskem jeziku C deklariraj polje 25-tih elementov strukture Tocka (elementi so X in Y, ki naj bosta celi števili in oblika, ki naj bo niz 20-tih znakov) in naredi naslednje naloge:
   1. S pomočjo kazalca na polje napolni tabelo z naključnimi vrednostmi, ki naj bodo:

-Za X interval [1,100)

-Za Y interval (0, 40]

-Za obliko pa niza »krog« oziroma »polni krog«.

* 1. Izpiši vse vrednosti tabele s pomočjo kazalca, tako da bodo izpisane najprej vse točke na lihih indeksih, nato pa še na sodih indeksih.
  2. Programu dodaj funkcijo float razdalja (short m, short n), ki izračuna razdaljo med točkama, ki ležita na m-tem in n-tem indeksu tabele. Poleg kazalca na polje uporabljal v funkciji še kazalce na spremenljivke, s katerimi računaš razdaljo med točkama. Prikaži klic te funkcije v glavnem programu in razdaljo izpiši na ekran.
  3. \*Programu dodaj funkcijo izriši, ki bo še izrisala vse točke tabele in njihove razdalje na ekran. Funkcija naj obvezno uporablja kazalce na polje, v funkciji lahko uporabljaš funkcije za delo z grafiko (knjižnica graphics.h ali druga).

1. V programskem jeziku C deklariraj strukturo »Racun« (Elementi so Ime in Priimek, ki naj bosta niza 15-tih znakov, ter ugnezdena struktura Stanje. Ugnezdena struktura Stanje naj vsebuje začetno\_stanje, ki naj bo realnega tipa ter polje DvigPolog, ki naj ima 100 realnih elementov.) in naredi naslednje naloge:
   1. S pomočjo kazalca na strukturo »Racun« napolni elemente te strukture z naključnimi vrednostmi, ki naj bodo:

-za Ime nizi »Janez«, »Polda« ali »Micka« ,

-za Priimek nizi »Novak«, »Medved« ali »Zajc«,

-začetno stanje naj ustreza intervalu (0..1000],

-tabela DvigPolog pa naj vsebuje števila z intervala [-3000..5000].

* 1. S pomočjo kazalca na strukturo »Racun« izpiši vse podatke o lastniku računa, tabela »DvigPolog«, pa naj bo izpisana tako, da bodo najprej izpisani vsi pologi na račun (pozitivna števila), nato pa še vsi dvigi (negativna števila).
  2. Programu dodaj funkcijo float stanje(struct Racun mojracun), ki izračuna, s pomočjo začetnega stanja ter polja »DvigPolog«, kakšno je trenutno stanje na računu. Funkcija naj uporablja kazalec na strukturo Racun ter kazalce na spremenljivke, s katerimi boš računal stanje na računu. Prikaži klic te funkcije v glavnem programu ter izpiši trenutno stanje računa na ekran.

1. V programu želimo hraniti podatke o računalniških igrah, katere elementi so: ime igre, vrsta igre (npr. simulacija, streljačina, ipd.) in število doseženih točk. Igralec (ta se deli na: uporabniško ime, geslo in IP naslov) lahko igra do 100 različnih računalniških iger in za vsako odigrano igro hranimo njegov najboljši dosežen rezultat (gre za celoštevilčne točke).

1. Napišite ustrezne podatkovne tipe v C/C++, ki bodo s pomočjo kazalca na tabelo omogočili hranjenje 50-ih igralcev
2. Napišite funkcijo v C/C++, ki bo s pomočjo kazalca na tabelo vrnila najboljši dosežen rezultat vseh igralcev pri prvoosebni streljačini z naslovom The Doom.
3. Napišite funkcijo v C/C++, ki bo razvrstila igre znotraj posameznih igralcev glede na njihovo vrsto (v naraščajočem vrstnem redu) ter bo izpisala tako urejen iger na ekran za tistega igralca, katerega uporabniško ime je enako nizu »C++ rulez!«. Tudi ta funkcija naj razvršča podatke obvezno z uporabo kazalca na polje, pri zamenjavi dveh iger pa naj obvezno uporablja še kazalec na spremenljivko (tmp).

Delovanje funkcij obvezno testirajte v glavnem programu! V ta namen lahko dodate programu tudi funkciji za vnos in izpis podatkov.

2. V programu želimo hraniti podatke o Računalnikih, ki naj vsebujejo: vrsto procesorja, takt procesorja, grafično kartico (znamka in model grafičnega čipa, velikost DDR3 pomnilnika) ter podatke o trdih diskih. Trdi disk je sestavljen iz: proizvajalca, kapacitete ter vrste diska (npr: HDD SATA2, SSD, itd.) Naj velja pravilo, da ima lahko posamezen računalnik vgrajeno le eno grafično kartico in 16 največ trdih diskov.

1. Napišite ustrezne podatkovne tipe v C/C++, ki bodo omogočili hranjenje do 100 računalnikov.
2. Napišite funkcijo v C/C++, ki s pomočjo kazalca na tabelo, za vse računalnike s procesorjem I7, izpisala skupno kapaciteto SSD diskov, za vsak računalnik posebej. Pri tem naj velja, da diski, ki so dejansko vgrajeni v računalnik imajo kapaciteto > 0, prazne reže za diske pa prepoznamo tako, da je kapaciteta = 0.
3. Napišite funkcijo v C/C++, ki bo razvrstila računalnike glede na vrsto procesorja (v padajočem vrstnem redu), v kolikor pa bosta računalnika imela enaka procesorja, pa naj funkcija sortira po vrsti grafičnega procesorja. Funkcija naj še izpise tako urejen seznam računalnikov na ekran. Tudi ta funkcija naj razvršča podatke obvezno z uporabo kazalca na polje, pri zamenjavi dveh iger pa naj obvezno uporablja še kazalec na spremenljivko (tmp).