**Kolegji Universitar “Bedër”**

**Fakulteti i Teknologjisë dhe Biznesit**

**Departamenti i Shkencave Kompjuterike**

**Semestri i Vjeshtës / 2023-2024**

**CMP111-1: Hyrje në Algoritme dhe Programim**



**Projekt Lëndor**

**Emri, Mbiemri: (Shkruani Emrin dhe Mbiemrin Tuaj)**

**Programi i Studimit: (Shkruani Programin Tuaj të Studimit)**

|  |  |
| --- | --- |
| **Numri i Pyetjeve të Zgjidhura** | **(Shkruani numrin e Pyetjeve që keni zgjidhur)** |
| **Pikët e Marra** |  |

**Vendosni zgjidhjen e çdo pyetjeje pas kërkesës së secilës prej tyre.**

**Pyetjet e Projektit**

1. Të shkruhet programi i cili numrin e sekondave të lexuar përmes tastierës e kthen në formatin HH:MM:SS.

#include <iostream>

int main() {

// Përcakto një numer të caktuar të sekondave (p.sh., 3665)

int totalSeconds = 3665;

// Kthehet në orë, minuta dhe sekonda

int hours = totalSeconds / 3600;

int minutes = (totalSeconds % 3600) / 60;

int seconds = totalSeconds % 60;

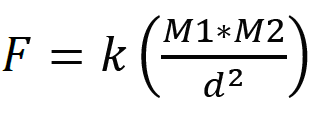
// Shfaq rezultatin

std::cout << "Koha e konvertuar: " << hours << ":" << minutes << ":" << seconds << std::endl;

return 0;

}

1. Ligji i Njutonit thotë që forca F midis dy trupave M1 dhe M2 është:



* ku **k** është konstanta e gravitetit dhe d është distanca ndërmjet
* trupave. Vlera e përafërt e konstantes **k** është 6.67 x 10-18 cm2/g2.

Të shkruhet programi për llogaritjen e forcës, nëse dihen masat e trupave dhe distanca në mes tyre.

#include <iostream>

int main() {

// Konstanta e gravitetit

const double k = 6.67e-18; // cm^2/g^2

// Masat e trupave (Masa M1 dhe Masa M2)

double M1 = 5.0; // g

double M2 = 3.0; // g

// Distanca ndërmjet trupave (d)

double d = 10.0; // cm

// Llogaritja e forcës përdorur Ligjin e Njutonit

double force = k \* (M1 \* M2) / (d \* d);

// Shfaq rezultatin

std::cout << "Forca midis trupave eshte: " << force << " dyn" << std::endl;

return 0;

}

1. Të shkruhet programi duke përdorur operatorët && dhe || përmes të cilit kontrollohet vlera hyrëse nga tastiera nëse është shkronjë apo jo, dhe shfaqet mesazhi perkatës.

#include <iostream>

int main() {

char input;

// Merr vlerën hyrëse nga tastiera

std::cout << "Shkruani nje shkronje: ";

std::cin >> input;

// Kontrollo nese vlera hyrese eshte shkronje ose jo

if ((input >= 'a' && input <= 'z') || (input >= 'A' && input <= 'Z')) {

std::cout << "Vlera hyrese eshte nje shkronje." << std::endl;

} else {

std::cout << "Vlera hyrese nuk eshte nje shkronje." << std::endl;

}

return 0;

}

1. Të shkruhet programi i cili cakton notën e studentit në varësi të pikëve të cilat

si vlerë hyrëse dhe shfaq mesazhin perkatës.

|  |  |
| --- | --- |
| **Segmenti i Pikeve** | **Nota** |
| 0 – 44 | 4 |
| 45 – 54 | 5 |
| 55 – 64 | 6 |
| 65 – 74 | 7 |
| 75 - 84 | 8 |
| 85 – 94 | 9 |
| 95 - 100 | 10 |
|  | Vlerë e panjohur |

#include <iostream>

int main() {

int piket;

// Merr vlerën hyrëse nga tastiera

std::cout << "Shkruani piket: ";

std::cin >> piket;

// Cakton notën bazuar në segmentin e pikëve

if (piket >= 0 && piket <= 44) {

std::cout << "Nota: 4" << std::endl;

} else if (piket >= 45 && piket <= 54) {

std::cout << "Nota: 5" << std::endl;

} else if (piket >= 55 && piket <= 64) {

std::cout << "Nota: 6" << std::endl;

} else if (piket >= 65 && piket <= 74) {

std::cout << "Nota: 7" << std::endl;

} else if (piket >= 75 && piket <= 84) {

std::cout << "Nota: 8" << std::endl;

} else if (piket >= 85 && piket <= 94) {

std::cout << "Nota: 9" << std::endl;

} else if (piket >= 95 && piket <= 100) {

std::cout << "Nota: 10" << std::endl;

} else {

std::cout << "Vlerë e panjohur" << std::endl;

}

return 0;

}

1. Të shkruhet programi për konvertimin e inçëve në centimetra. Vlera e inçëve të jepet përmes tastierës (1 inç=2.54 cm). Kahu i konvertimit të jetë me zgjedhje.

#include <iostream>

int main() {

double inches, centimeters;

// Merr vlerën hyrëse nga tastiera

std::cout << "Shkruani gjatesine ne inca: ";

std::cin >> inches;

// Kthehet gjatesia ne centimetra

centimeters = inches \* 2.54;

// Shfaq rezultatin

std::cout << inches << " inca eshte " << centimeters << " centimetra." << std::endl;

return 0;

}

1. Të shkruhet programi i cili llogarit pagën NETO (pn) nëse dihet paga BRUTO (pb).
   1. Punëtorit i ndalen 5% të pagës bruto për trustin pensional (pen), kurse 5% i paguan punëdhënësi.
   2. Punëtorit i ndalet tatimi në pagë (tat) e i cili është i përshkallëzuar në varësi të pagës bruto dhe llogaritet pasi të jetë hequr pagesa për trustin pensional.

|  |  |
| --- | --- |
| **Segment i Pagesës Bruto** | **Përqindja e Tatimit** |
| 0 < pb <= 30000 | 0% |
| 30001 < pb <= 40000 | 4% |
| 40001 < pb <= 55000 | 8% |
| 55001 < pb | 10% |

#include <iostream>

int main() {

double pb, pn, pen, tat;

// Merr vlerën hyrëse nga tastiera

std::cout << "Shkruani pagën BRUTO: ";

std::cin >> pb;

// Llogaritja e pagesës për trustin pensional

pen = 0.05 \* pb;

// Pagesa NETO pasi të jetë hequr pagesa për trustin pensional

pn = pb - pen;

// Llogaritja e tatimit në pagë

if (pb > 0 && pb <= 30000) {

tat = 0.0;

} else if (pb > 30000 && pb <= 40000) {

tat = 0.04 \* pn;

} else if (pb > 40000 && pb <= 55000) {

tat = 0.08 \* pn;

} else if (pb > 55000) {

tat = 0.10 \* pn;

}

// Pagesa NETO pasi të jetë hequr edhe tatimi

pn -= tat;

// Shfaq rezultatin

std::cout << "Paga NETO e punetorit eshte: " << pn << std::endl;

return 0;

}

1. Të shkruhet programin për makinen llogaritëse të thjeshtë duke përdorur switch e i cili kryen llogaritje për dy numra të plotë. Nëse shtypet + të kryhet mbledhja, nëse shtypet - të kruhet zbritja, nëse shtypet / të kryhet pjesëtimi, nëse shtypet \* të kryhet shumëzimi, për karaktere tjera të shfaqet mesazhi "Gabim në llogaritje".

#include <iostream>

int main() {

char operacioni;

int numri1, numri2, rezultati;

// Merr vlerat hyrëse nga tastiera

std::cout << "Shkruani numrin e pare: ";

std::cin >> numri1;

std::cout << "Shkruani numrin e dyte: ";

std::cin >> numri2;

std::cout << "Zgjedhni operacionin (+, -, \*, /): ";

std::cin >> operacioni;

// Kryej llogaritjen bazuar në operacionin zgjedhur

switch (operacioni) {

case '+':

rezultati = numri1 + numri2;

break;

case '-':

rezultati = numri1 - numri2;

break;

case '\*':

rezultati = numri1 \* numri2;

break;

case '/':

// Kontrollojë pjesëtimin me zero

if (numri2 != 0) {

rezultati = numri1 / numri2;

} else {

std::cout << "Gabim: Pjesëtimi me zero nuk është i lejuar." << std::endl;

return 1; // Dil nga programi në rastin e gabimit

}

break;

default:

std::cout << "Gabim në llogaritje: Operacioni i papranuar." << std::endl;

return 1; // Dil nga programi në rastin e gabimit

}

// Shfaq rezultatin

std::cout << "Rezultati: " << rezultati << std::endl;

return 0;

}

1. Të shkruhet programi për mbledhjen, zbritjen, shumëzimin dhe pjesëtimin e dy numrave të plotë:

#include <iostream>

int main() {

int numri1, numri2;

// Merr vlerat hyrëse nga tastiera

std::cout << "Shkruani numrin e pare: ";

std::cin >> numri1;

std::cout << "Shkruani numrin e dyte: ";

std::cin >> numri2;

// Mbledhja

int mbledhja = numri1 + numri2;

std::cout << "Mbledhja: " << mbledhja << std::endl;

// Zbritja

int zbritja = numri1 - numri2;

std::cout << "Zbritja: " << zbritja << std::endl;

// Shumëzimi

int shumezimi = numri1 \* numri2;

std::cout << "Shumezimi: " << shumezimi << std::endl;

// Kontrollojë pjesëtimin me zero

if (numri2 != 0) {

// Pjesëtimi

double pjesetimi = static\_cast<double>(numri1) / numri2;

std::cout << "Pjesetimi: " << pjesetimi << std::endl;

} else {

std::cout << "Gabim: Pjesëtimi me zero nuk është i lejuar." << std::endl;

return 1; // Dil nga programi në rastin e gabimit

}

return 0;

}

1. Të shkruhet programi për shfaqjen e karaktereve të alfabetit amerikan në formë trekëndëshi si në pamjen e mëposhtme duke e përdorur strukturën ciklike while.

A

B B

C C C

D D D D

E E E E E

#include <iostream>

int main() {

char karakteri = 'A';

int numri\_rreshtave = 5;

int rreshti = 1;

while (rreshti <= numri\_rreshtave) {

int numri\_karaktereve = rreshti;

// Shtyp karakteret për rreshtin aktual

while (numri\_karaktereve > 0) {

std::cout << karakteri << " ";

numri\_karaktereve--;

}

// Kalon në radhën e karakterit të ardhshëm

karakteri++;

// Kalon në rreshtin e ardhshëm

std::cout << std::endl;

rreshti++;

}

return 0;

}

1. Të shkruhet programi për konvertimin e numrit nga sistemi decimal në sistemin oktal

#include <iostream>

int main() {

int numri\_decimal, numri\_oktal = 0, pozita = 1;

// Merr vlerën hyrëse nga tastiera

std::cout << "Shkruani numrin decimal: ";

std::cin >> numri\_decimal;

// Kthehet numri nga decimal në oktal

while (numri\_decimal > 0) {

numri\_oktal += (numri\_decimal % 8) \* pozita;

numri\_decimal /= 8;

pozita \*= 10;

}

// Shfaq rezultatin

std::cout << "Numri ne sistem oktal: " << numri\_oktal << std::endl;

return 0;

}

1. Të shkruhet programi për shfaqjen e piramidës të ndërtuar me ylla (\*) si në pamje. Numri i rreshtave të përcaktohet përmes tastierës

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \*\*

\* \* \* \* \* \* \* \* \*

#include <iostream>

int main() {

int numri\_rreshtave;

// Merr vlerën hyrëse nga tastiera

std::cout << "Shkruani numrin e rreshtave te piramides: ";

std::cin >> numri\_rreshtave;

// Krijon piramidën me ylla

for (int i = 1; i <= numri\_rreshtave; i++) {

// Shtyp yllat për rreshtin aktual

for (int j = 1; j <= 2 \* i - 1; j++) {

std::cout << "\* ";

}

// Kalon në rreshtin e ardhshëm

std::cout << std::endl;

}

return 0;

}

1. Të shkruhet programi për shfaqjen e tabelës së shumëzimit nga 1 deri në 10 në formë tabelare.

#include <iostream>

int main() {

// Krijon tabelën e shumëzimit

for (int i = 1; i <= 10; i++) {

for (int j = 1; j <= 10; j++) {

std::cout << i \* j << "\t";

}

std::cout << std::endl;

}

return 0;

}

1. Të shkruhet programi i cili mbush vektorin me 8 anëtarë. Mbushja të realizohet me strukturën ciklike while, kurse vlerat e anëtarëve të mbushen me kubin e indekseve. Shfaqja të bëhet me strukturën ciklike for.

#include <iostream>

int main() {

const int madhesia\_vektorit = 8;

int vektori[madhesia\_vektorit];

// Mbush vektorin me kubin e indekseve përmes while

int indeksi = 0;

while (indeksi < madhesia\_vektorit) {

vektori[indeksi] = indeksi \* indeksi \* indeksi;

indeksi++;

}

// Shfaq vektorin përmes for

for (int i = 0; i < madhesia\_vektorit; i++) {

std::cout << vektori[i] << " ";

}

return 0;

}

1. Të deklarohet dhe të mbushet **vektori[5]** me katrorët e indekseve. Më pas të kërkohet për vlerën 4 nëse ekziston. Nëse gjendet dhe të shfaqet pozicioni ku është gjetur në të kundërtën të shfaqet mesazhi se nuk është gjetur.

#include <iostream>

int main() {

const int madhesia\_vektorit = 5;

int vektori[madhesia\_vektorit] = {1, 4, 9, 16, 25};

// Kërkohet vlera 4 në vektor

int kerkuar = 4;

bool gjendet = false;

int pozicioni;

for (int i = 0; i < madhesia\_vektorit; i++) {

if (vektori[i] == kerkuar) {

gjendet = true;

pozicioni = i;

break;

}

}

// Shfaq rezultatin

if (gjendet) {

std::cout << "Vlera 4 gjendet në pozicionin " << pozicioni << std::endl;

} else {

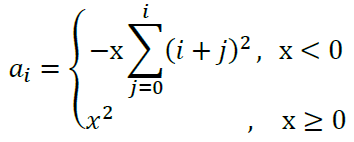
std::cout << "Vlera 4 nuk është gjetur në vektor." << std::endl;

}

return 0;

}

1. Të formohet vektori A(n) duke llogaritur anëtarët ai përmes shprehjes së mëposhtme nëse dihet vlera x dhe n.



1. Të formohet vektori C(m+2+n) nga anëtarët e vektorit A[m], vektorit me anëtar 0 dhe anëtarët e vektori B[n]. Formimi i vektorit të bëhet sipas modelit të mëposhtëm, nëse A[m]={10,20,30,40} dhe B[n]={-1,-2,-3,-4}.



1. Formoni një program që lexon vlerat e vektorit me n elemente dhe shfaq vlerat dhe indekset e elemetit më të vogël dhe më të madh.

#include <iostream>

#include <climits>

int main() {

const int n = 5; // Ndryshojeni sipas nevojës

int vektori[n];

// Lexon vlerat e vektorit nga tastiera

std::cout << "Shkruani vlerat e vektorit (" << n << " elemente): ";

for (int i = 0; i < n; i++) {

std::cin >> vektori[i];

}

// Gjen vlerën maksimale dhe indeksin e saj

int maksimumi = INT\_MIN;

int indeksi\_maks = -1;

// Gjen vlerën minimale dhe indeksin e saj

int minimumi = INT\_MAX;

int indeksi\_min = -1;

for (int i = 0; i < n; i++) {

if (vektori[i] > maksimumi) {

maksimumi = vektori[i];

indeksi\_maks = i;

}

if (vektori[i] < minimumi) {

minimumi = vektori[i];

indeksi\_min = i;

}

}

// Shfaq rezultatin

std::cout << "Vlera maksimale eshte: " << maksimumi << " ne indeksin " << indeksi\_maks << std::endl;

std::cout << "Vlera minimale eshte: " << minimumi << " ne indeksin " << indeksi\_min << std::endl;

return 0;

}

1. Të shkruhet funksioni perfekt i cili përcakton nëse parametri i tij është numër perfekt. (Numri është perfekt nëse shuma e faktorëve/plotpjesëtuesve të tij duke përfshirë edhe 1, por jo vetveten është baras me numrin. p.sh 6=1+2+3). Duke shfrytëzuar funksionin perfekt të shtypen të gjithë numrat perfekt nga 1 deri në 5000.

#include <iostream>

bool eshte\_perfekt(int numri) {

int shuma = 1;

for (int i = 2; i <= numri / 2; i++) {

if (numri % i == 0) {

shuma += i;

}

}

return (shuma == numri);

}

int main() {

std::cout << "Numrat perfekt nga 1 deri ne 5000 jane: ";

for (int i = 1; i <= 5000; i++) {

if (eshte\_perfekt(i)) {

std::cout << i << " ";

}

}

return 0;

}

1. Të shkruhen 3 funksione me emrin siperfaqja() të cilat llogarisin sipërfaqen e kubit, sferës dhe konit. Funksionet të thirren për vlera të lexuara përmes tastierës.

#include <iostream>

#include <cmath>

// Funksioni për llogaritjen e sipërfaqes së kubit

double siperfaqja\_kubit(double brinja) {

return 6 \* brinja \* brinja;

}

// Funksioni për llogaritjen e sipërfaqes së sferës

double siperfaqja\_sfera(double rrezja) {

return 4 \* M\_PI \* rrezja \* rrezja;

}

// Funksioni për llogaritjen e sipërfaqes së konit

double siperfaqja\_koni(double rrezja, double gjatesia) {

return M\_PI \* rrezja \* (rrezja + sqrt(gjatesia \* gjatesia + rrezja \* rrezja));

}

int main() {

double brinja\_kub, rrezja\_sfera, rrezja\_koni, gjatesia\_koni;

// Lexon vlerat e brinjës, rrezjës dhe gjatësisë nga tastiera

std::cout << "Shkruani brinjen e kubit: ";

std::cin >> brinja\_kub;

std::cout << "Shkruani rrezjen e sferes: ";

std::cin >> rrezja\_sfera;

std::cout << "Shkruani rrezjen e konit: ";

std::cin >> rrezja\_koni;

std::cout << "Shkruani gjatesine e konit: ";

std::cin >> gjatesia\_koni;

// Thirr funksionet për llogaritjen e sipërfaqeve dhe shfaq rezultatin

std::cout << "Siperfaqja e kubit eshte: " << siperfaqja\_kubit(brinja\_kub) << std::endl;

std::cout << "Siperfaqja e sferes eshte: " << siperfaqja\_sfera(rrezja\_sfera) << std::endl;

std::cout << "Siperfaqja e konit eshte: " << siperfaqja\_koni(rrezja\_koni, gjatesia\_koni) << std::endl;

return 0;

}

1. Të shkruhet funksioni shumezimi i cili gjeneron në mënyrë të rastësishme dy numra nga 0-10 dhe shfaq për prodhimin në mes e tyre.
2. Të shkruhet programi i cili përmes funksionit aritmetika kryen veprimet e mbledhjes, zbritjes, shumëzimit dhe pjesëtimit për dy numra të gjeneruar në mënyrë të rastësishme. Përmes funksionit menyja të pyetet për veprimin që duhet të kryhet (mbledhje, zbritje, shumëzim, pjesëtim apo kombinim i rastësishëm).
3. Të shkruhet programi në C++ për krijimin dhe mbushjen e matricës katrore A(m,m), nëse m=5. Anëtarët e matricës të mbushen sipas shprehjes aij=i.

#include <iostream>

int main() {

const int m = 5; // Ndryshoni sipas nevojës

int matrica[m][m];

// Mbushja e matricës sipas shprehjes aij=i

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

matrica[i][j] = i + 1;

}

}

// Shfaqja e matricës

std::cout << "Matrica e krijuar:\n";

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

std::cout << matrica[i][j] << "\t";

}

std::cout << std::endl;

}

return 0;

}

1. Të shkruhet programi në C++ për krijimin dhe mbushjen e matricës katrore A(m,m), nëse m=3. Mbushja të realizohet përmes strukturës ciklike while, kurse shfaqja e anëtarëve të realizohet përmes strukturës ciklike do-while. Anëtarët e matricës të jepen përmes tastierës.

#include <iostream>

int main() {

const int m = 3; // Ndryshoni sipas nevojës

int matrica[m][m];

// Mbushja e matricës përmes strukturës ciklike while

int i = 0, j = 0;

while (i < m) {

while (j < m) {

std::cout << "Shkruani elementin ne rreshtin " << i + 1 << " dhe kolonën " << j + 1 << ": ";

std::cin >> matrica[i][j];

j++;

}

j = 0; // Rifillon për një rresht të ri

i++;

}

// Shfaqja e matricës përmes strukturës ciklike do-while

i = 0;

do {

j = 0;

do {

std::cout << matrica[i][j] << "\t";

j++;

} while (j < m);

std::cout << std::endl;

i++;

} while (i < m);

return 0;

}

1. Të shkruhet programi në C++ për mbushjen e matricës A(m,m), nëse m=6. Pjesa e brendshme e matricës të mbushet me 0, kurse korniza e matricës të mbushet me 1.
2. Të formohen matrica A(m,m) e cila përmban notat e studentëve. Të llogaritet nota mesatare e secilit student dhe ajo të ruhet në vektorin B(m,n).

