

**KODI A.IV.3.2 DETYRË KURSI: ESE/KOMENTAR LIBRI/DETYRË NË GRUP/PJESËMARRJE NË  
KËRKIM SHKENCOR**

**FAKULTETI:** Inxhinierisë, Informatikës dhe Arkitekturës  
**LËNDA:** Algoritmikë dhe Hyrje në Programim  
**PEDAGOGU:** Dr. Ardiana Topi, MSc. Jugerta Gurabardhi  
**DATA E ZHVILLIMIT:** 08.02.2023  
**EMRAT E STUDENTËVE:** Daniel Hudhra, Arsid Baculi, Gjergj Cumraku  
**PROGRAMI:** Inxhinieri Informatike

**Tema e detyrës:**

**3. Ndërtimi i një projekti në C++ që bën kontrollin e vlefshërisë së kartave të bankës.**

Në këtë sistem, pas futjes së numrit të kartës së kreditit, sistemi verifikon vlefshmërinë e saj dhe përcakton nëse bëhet fjalë për një kartë Visa, MasterCard, American Express apo një formë tjetër karte. Ai verifikon origjinalitetin e kartës dhe llojin e kartës së kreditit duke kryer disa operacione dhe verifikime bazë. Numrat e kartës së kreditit të futur nga përdoruesi gjenerohen së bashku me një mesazh që tregon nëse karta është e vlefshme apo e pavlefshme. Nëse numri i kartës së kreditit të futur është i vlefshëm, do të printohet edhe lloji i kartës së kreditit.

Si pjesë e kësaj detyre, studenti duhet të bëjë analizën e problemit, ndërtimin e diagramës dhe zhvillimin e programit në gjuhën C++.

**I. HYRJE/PARASHTRIMI I TEZËS**

Të gjithë e dimë që kartat e kreditit janë një mundësi e mirë për të blerë të mira (mallra apo shërbime), mundësojnë oferta (zbritje), si dhe lehtësojnë procesin e blerjeve. Për më tepër, ato mund të përdoren edhe si llogari rrjedhëse, nga ku mund të tërheqim para p.sh nga llogaria jonë e pagës, por ama jo nga paratë e bankës. Shumë kompani në tregun financiar ofrojnë karta krediti.

Krahas të mirave, duhet t'i kushtojmë rëndësi sigurisë së të dhënave personale, kështu mund t'i lëmë "feedback" kompanive për siguri më të lartë ndaj sulmeve të hacker-ave ndaj bizneseve për parandalimin e Phishing dhe Spoofing. Këto dy mënyra nënkuptojnë përdorimin e email-eve të rremë ose shtirjen me qëllim vjedhjen e informacionit.

Nga pikëpamja e **programimit**, përdorimi i programeve kompjuterike lidhur me kartat e kreditit ka perspektivë në pjesën e website-eve dhe aplikacioneve të ndryshme që aplikojnë mundësinë e përdorimit të kartave të kreditit, si dhe përcaktimin e vlefshmërisë dhe llojit të tyre.

Tani për tani, synimet tona për këtë detyrë janë:

- Të shfrytëzojmë njohuritë e marra për të kuptuar si funksionon një sistem i thjeshtë duke dhënë shpjegime dhe sqarime për hapat e interpretuesit, rrjedhën dhe konvertimin e të dhënave.
- Të shkruajmë kod sa më të pastër (dry), duke iu bindur "Don't repeat yourself principle".

- Të përdorim emra variablash dhe funksionesh sa më përshkruese.
- Të evitojmë gabimet sintaksore dhe logjike, si edhe shmangien e cikleve të pafundme.

Në të ardhmen, presim ta zhvillojmë këtë program duke e bërë më kompleks përmes shtimit të përdorimit të modularizimit, si dhe të funksionaliteteve të menuesë duke u frymëzuar nga programe të ngjashme si “Credit Score Simulator” në web.

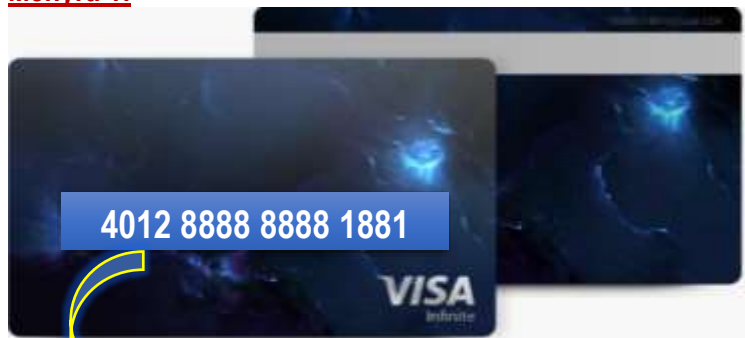
## II. ZHVILLIMI/ARGUMENTIMI

### 1. Analiza e problemit:

#### 🌀 Logjika e kontrollit të vlefshmërisë së një karte 🌀

- 🔵 Sipas hulumtimeve që bëmë, kontrolli i vlefshmërisë të këtyre kartave realizohet përmes algoritmit “checksum” të Hans Peter Luhn-it, të cilin po e argumentojmë me dy mënyra.
- 🔵 Paraprakisht, po marrim në shqyrtim të njëjtën kartë Visa të vlefshme për të dyja rastet:

#### Mënyra 1:



4012 8888 8888 1881

VISA

Infinite

E përmbysim

<b>Shifra</b>	4	0	1	2	8	8	8	8	8	8	8	8	1	8	8	1
<b>×2</b>	8	0	2	2	16	8	16	8	16	8	16	8	2	8	16	1
<b>×2-9</b>	8	0	2	2	7	8	7	8	7	8	7	8	2	8	7	1
<b>Shuma</b>	1+7+8+2+8+7+8+7+8+7+8+7+2+2+0+8 = 90 ⇔ 90 mod 10 = 0 (e vlefshme)															

**Hapi 1:** Nga shifra më në të djathtë, dyfishojmë vlerën e çdo elementi (shifre) dy e nga dy.

**Hapi 2:** Nëse shifrat e dyfishuara janë më të mëdhaja se 9 (p.sh në rastin tonë  $8 \times 2 > 9$ ), zbriti 9 ( $16-9=7$ ).

**Hapi 3:** Mbledhim të gjitha shifrat e përfutuara.

**Hapi 4:** Karta është e vlefshme në rast se mbetja e shumës me 10 është 0 (shuma mod 10 == 0).

#### Mënyra 2:

<b>Shifra</b>	4	0	1	2	8	8	8	8	8	8	8	8	1	8	8	1
---------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Shuma e vendeve tek (Djathta → Majtë):  $1+8+8+8+8+8+2+0 = 43$

"Çift"	4		1		8		8		8		8		1		8	
×2	8		2		16 => 1+6		16 => 1+6		16 => 1+6		16 => 1+6		2		16 => 1+6	

Shuma e vendeve çift të përfutur më sipër:  $8+2+7+7+7+7+2+7 = 47$

**Shuma** Mbledh dy shumat e vendeve çift & tek përkatëse:  
 $43+47=90 \Leftrightarrow 90 \bmod 10 = 0$  (e vlefshme)

Si përfundim, vërejmë që me të dyja metodat, shifra në fund na del po njësoj (90), e meqenëse 90 plotpjesëtohet me 10 (që mbetja del 0), atëherë karta është e vlefshme.

**Meqë të dyja mënyrat janë funksionale, po zgjedhim të punojmë me të parën.**



Lind pyetja, si do ishte zgjidhja më efiçente e shfrytëzimit të logjikës së këtij algoritmi për ta implementuar në kod (për përcaktimin e vlefshmërisë së kartës)?

Mendojmë që gjithë numrat e kartës që do fusë përdoruesi t'i ruajmë në një string. P.sh po marrim rastin nëse përdoruesi fut kartën "Visa" 16-shifrore: **4012888888881881**

Këto vargje karakteresh trashëgohen nga gjuha C. Në fakt, ky varg përfundon me karakterin null. Në rastin konkret, kartën mund ta mendojmë si koleksion prej 17 karakteresh:

'4', '0', '1', '2', '8', '8', '8', '8', '8', '8', '8', '8', '1', '8', '8', '1' dhe '\0'.

Ky i fundit përcakton një karakter të pavlefshëm (terminatorin).

Kështu, e kemi më të thjeshtë për zbatimin e algoritmit të Luhn-it, si dhe mund të shfrytëzojmë njohuritë e vektorëve:

- 1) Në bazë të indekseve të karaktereve (për të iteruar në to) përmes ciklit "For".
- 2) Për ta përmbysur (reverse) string-un duke filluar nga shifra nga më në të djathtë, duke e ruajtur në një integer (në mënyrë që të kryejmë veprime matematikore).
- 3) Për të shumëzuar dy e nga dy çdo shifër.
- 4) Nëse shifrat e dyfishuara janë më të mëdha se 9, zbriti 9.
- 5) Nëse mbetja me 10 e shumës (së të gjitha shifrave të përfutura) është 0, karta është e vlefshme. Përndryshe, nuk është e vlefshme.
- 6) Marrim dy shifrat e para për kontrollin e llojit të kartës (përmes funksionit të gatshëm substr).

### Formati i kartave të kreditit

Për kontrollin e **formatit të kartës së kreditit** (nëse është Visa, AmericanExpress, MasterCard etj) mendojmë të shfrytëzojmë shprehjen e shumëfishtë if – else if – else.  
Do i referohemi tabelës së mëposhtme:

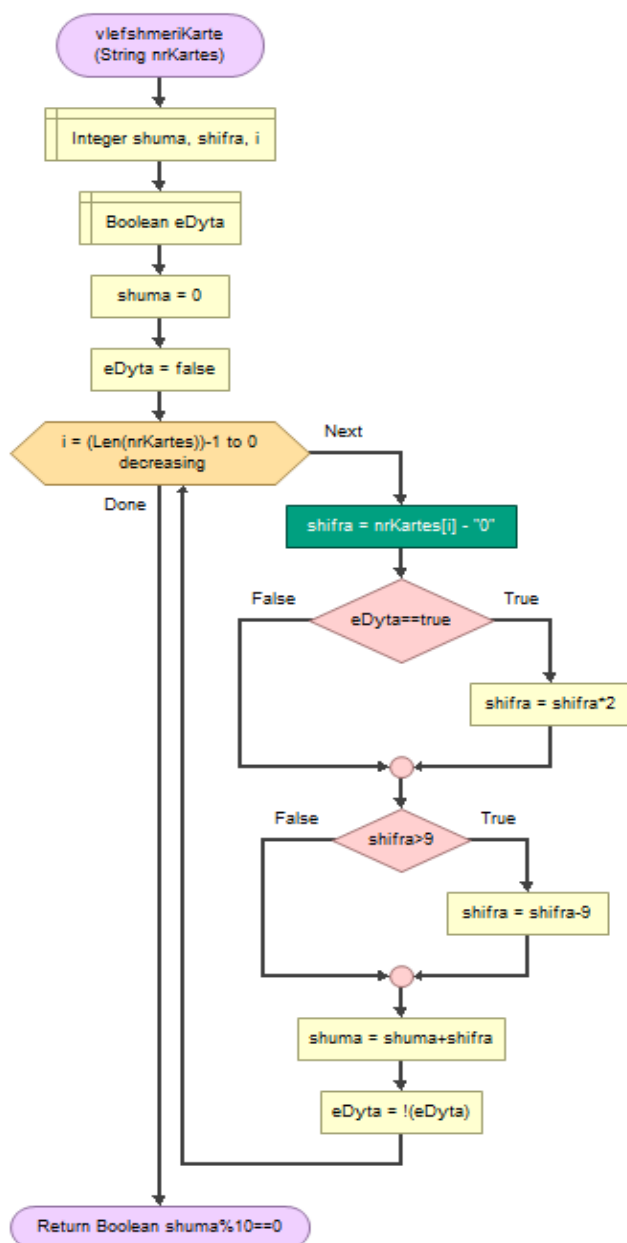
Lloji:	Fillon me:	Shifra:
American Express	34 ose 37	15
Visa	4	13 ose 16
MasterCard	51-55	16

Discover	6011 ose 5	16 ose 15 (përkatësisht)
Diners Club	300-305, 36 ose 38	14
JCB	2131 ose 1800	15

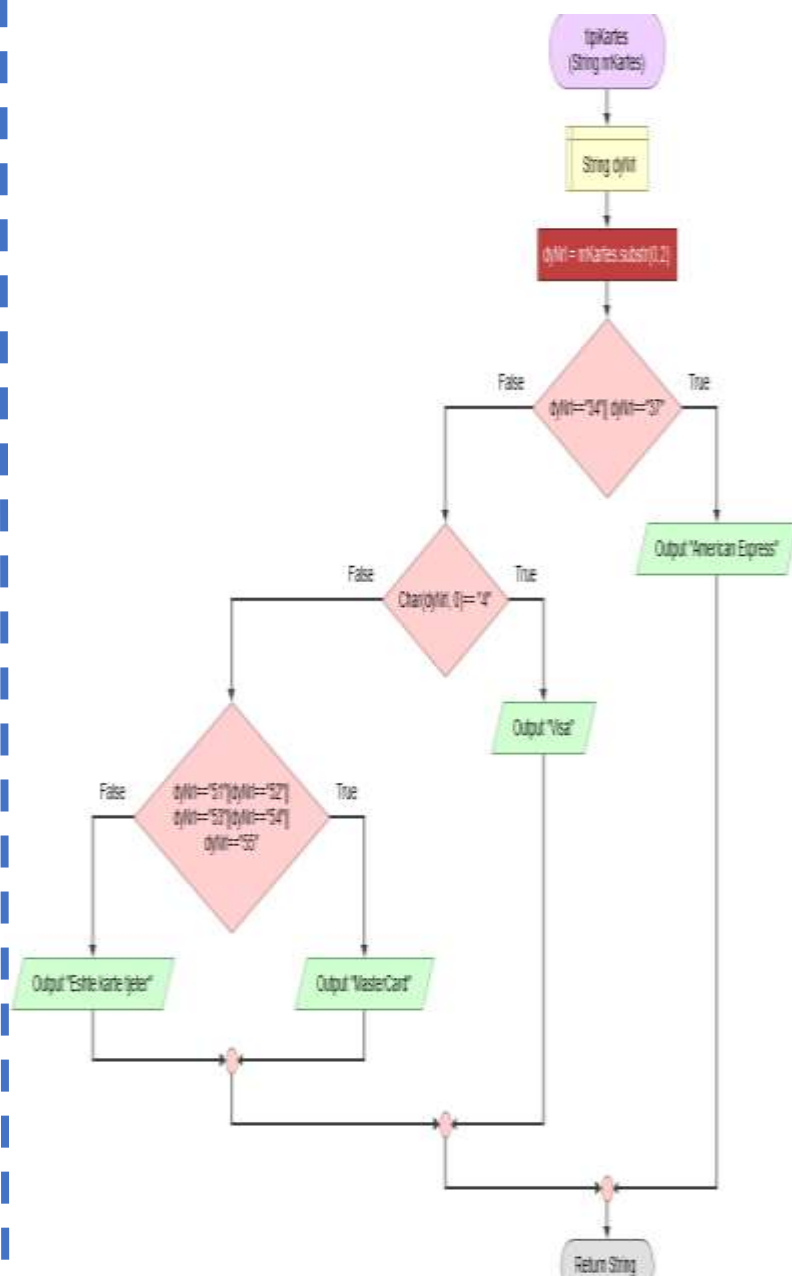
## 2. Ndërtimi i diagramës:

Duke rikujtuar që logjika e programit është nga lart→poshtë dhe majtas→djathtas, atëherë po ndërtojmë diagramën e rrjedhës së të dhënave, pra blokskemat (flowcharts). Po përdorim programin **Flowgorithm**.

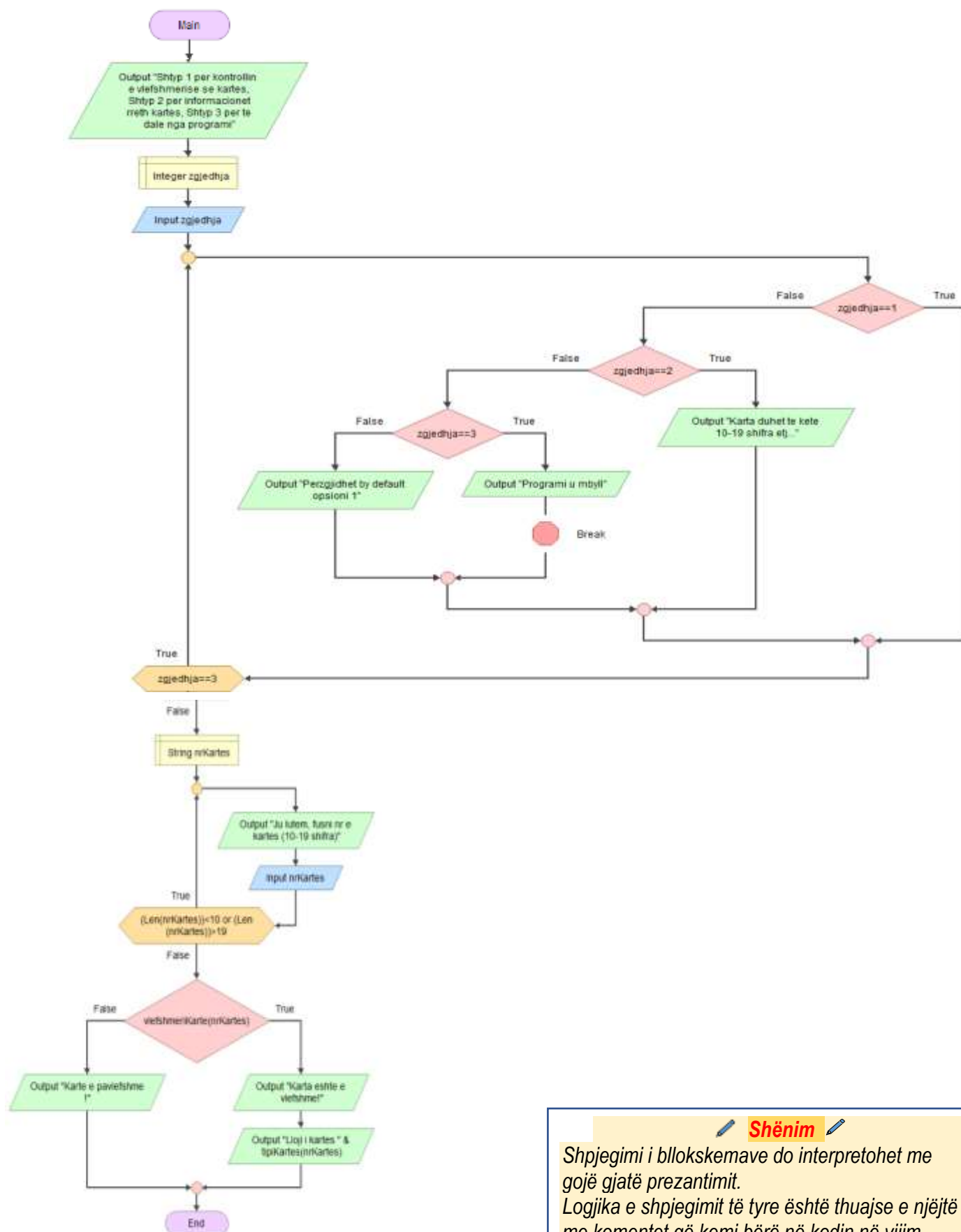
### Blokskema për funksionin "vlefshmeriKarte"



### Blokskema për funksionin "vlefshmeriKarte"

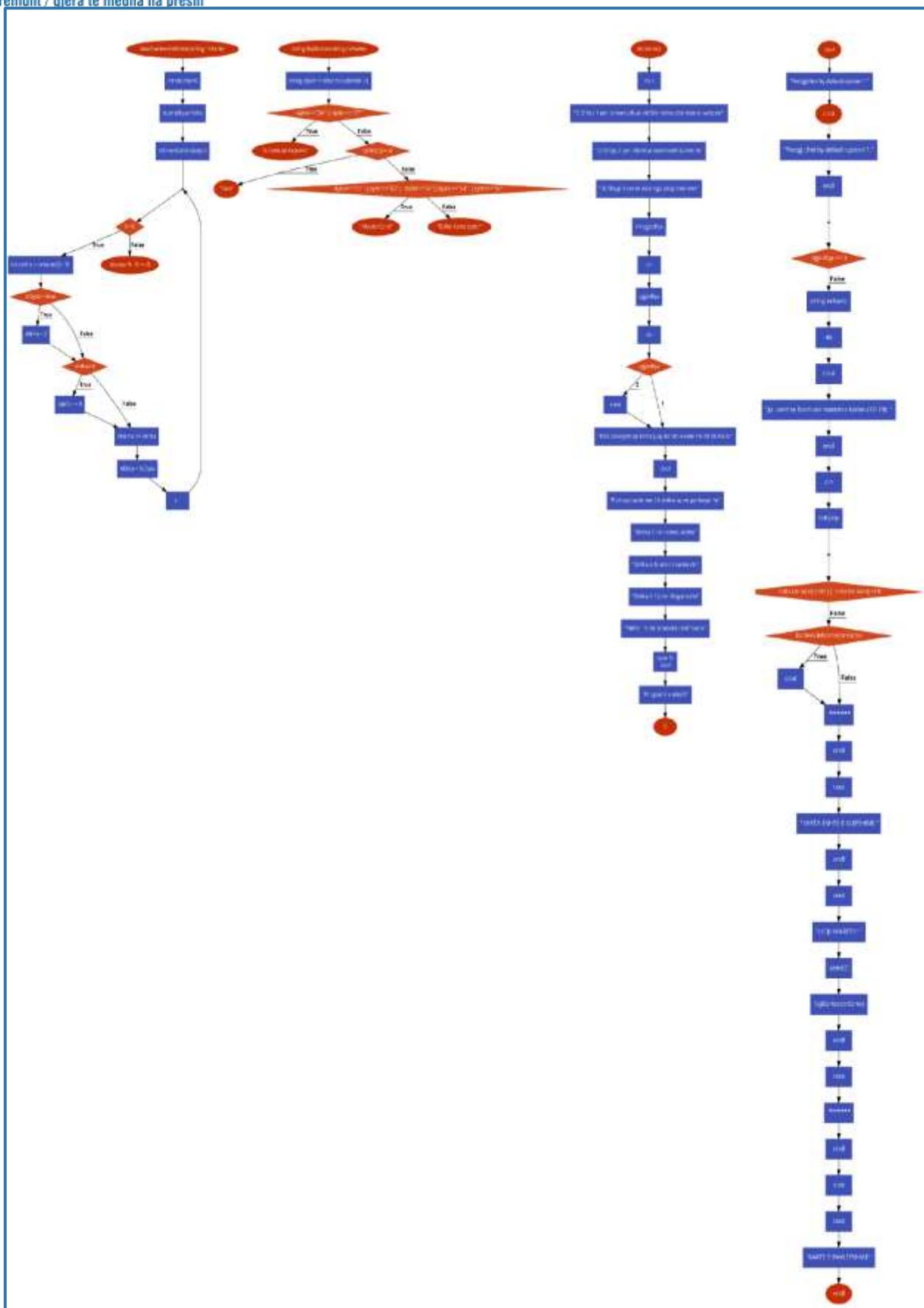


Blokskema për funksionin "Main"



**Shënim**

Shpjegimi i bllokskemave do interpretohet me gojë gjatë prezantimit.  
 Logjika e shpjegimit të tyre është thuajse e njëjtë me komentet që kemi bërë në kodin në vijim.



Kjo është një bllokskemë më e përmblodhur e të trija funksioneve që kemi ndërtuar, por e ndërtuar me një program tjetër.

### 3. Zhvillimi i programit në gjuhën C++:

Më poshtë, kemi zhvilluar argumentimin dhe shpjegimin e detajuar të kodit me anë të komenteve (me ngjyrë të kuqe).

```
#include <iostream>    //direktivë paraprocesorësh
#include <string>       //për përdorimin e klasës string, përfshijmë header file-in <string>
#include <iomanip>       //kemi mësuar që kjo librari mundëson funksionin setw
using namespace std;   //(standard library), shmang përdorimin e std:: para objekteve cin/cout
```

```
/* -----SHEMBUJ PER TESTIM-----
VISA:      4111111111111111  4012888888881881
MasterCard: 555555555554444
American Express: 378282246310005
Discover:   6011111111111117
*/
```

// 1) Funkcioni që kontrollon vlefshmërinë e kartës duke u bazuar te algoritmi i Luhn-it

```
/*
5Fillojmë me përcaktimin (krijimin) e një funksionit me emrin "karteevlefshme", tipi i kthimit i të cilit është boolean (vlerë T/F). Si parametër formal do shërbejë një string (nrKarte). Në trupin e funksionit, do arsyetojmë më poshtë duke implementuar në kod algoritmin e Luhn-it që trajtuam më sipër.
//(Pra, logjika e këtij funksioni kontrollon nëse numri i kartës së kreditit është i vlefshëm.)
*/
```

```
bool karteevlefshme(string nrKarte) {
    int shuma=0;    //deklaroj & inicializoj me 0 variablin "shuma" (nr. i plotë)
    bool eDyta=false; //variabli i tipit boolean (shërben si mekanizëm për marrjen dy e
                    //nga dy të shifrave (më poshtë)
    /*
```

Përderisa e dimë sa herë do përsëritet cikli, përdorim atë "For". Supozojmë se futet një string 16-shifror nga përdoruesi. Duke filluar nga gjatësia e string-ut ( 16 – që llogaritet nga funksioni size() ) minus 1 (pra 15), pyes/testoj nëse "a është i>=0"? Nëse kushti është True, trupi i ciklit ekzekutohet dhe i dekrementohej në 14. E njëjta logjikë vlen dhe për iteracionet e tjera derisa i=0. Në total, cikli përsëritet 16 herë.

```
*/
```

```
//Për referim, po marr një kartë të vlefshme "Visa":
```

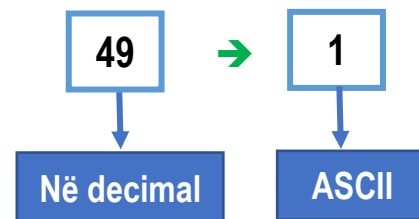
4012888888881881

```
for (int i=nrKarte.size()-1; i>=0; i--) {
```

/\* Te variabli shifra do kopjohet iteracion pas iteracioni vlera e vektorit me karaktere (me emrin nrKarte) duke filluar nga indeksi i fundit.



Në iteracionin e parë, do merret vlera e indeksit të fundit të vektorit që korrespondon me shifrën e fundit (1). Pra, `nrKarte[15] = 1`. Mirëpo, nëse nuk i zbresim '0' (karakter), atëherë ky karakteri 1 (i fundit), do i ruhet vetëm vlera përkatëse në decimal. Pra, për ta kthyer në vlerë ASCII (49-ën):  $49 - '0' = 49 - 48 = 1$



E njëjta logjikë vlen dhe për shifrat e tjera deri në `nrKarte[0]`. Me pak fjalë, ne bëmë reverse (përmbysem) string-un apo kartën e futur nga përdoruesi.

```

*/
int shifra = nrKarte[i] - '0';
if (eDyta==true) //Pyes, a është boolean-i eDyta == true? Në iteracionin 1 jo.
    shifra *= 2; //Në iteracionin 2, 4, 6 etj, kopjo te variabli shifra vlerën e po këtij variabli
                //shumëzuar me 2 (pra thjesht dyfishojmë dy e nga dy).
if (shifra>9) //Pyes, a është variabli shifra > 9?
    shifra -= 9; //Nëse shifra e dyfishuar e plotëson kushtin, dekrementoje me 9.

```

```

shuma += shifra;

```

/\* Variabli `shuma` shërben si akumulues i shumës së të gjitha shifrave të përftuara. Pra, fillojmë që nga `shuma=0` ;  
`shuma=shuma+shifra=0+1=1 ...` e kështu me radhë. Duke shtuar `cout<<shuma<<endl`; do marrim rezultatet e pritura.

```

Ju lutem, fusni numrin e kartes:
4012888888881881
1
8
16
18
26
33
41
48
56
63
71
78
80
82
82
90

```

```

eDyta = !eDyta;
/* Me anë të operatorit logjik NOT, kopjojmë te variabli eDyta, vlerën e të kundërit të tij, në mënyrë që
çdo përsëritje variabli të kthehet në True për të shumëzuar me 2 shifrat dy e nga dy (çdo 1 në 2 shifra):
eDyta = !false = true ; eDyta = !true = false etj... */
}

```

```

return(shuma % 10 == 0); // Funkcioni kthen true vetëm nëse mbetja e
                        // variablit shuma me 10 është 0.

```

```

}

```

## // 2) Funkcioni që kontrollon llojin e kartës së kreditit

```

string llojiKartes(string nrKarte){ //tipi i kthimit = string

```

```

    string dyNrl = nrKarte.substr(0, 2); //përdorim funksionin substr për të bërë substring në
    //pozicionin 0 dhe me gjatësi 2 (që përkon me dy numrat e parë të kartës) dhe i ruajmë te variabli dyNrl

```

//Përdorim një strukturë if-else if-else të shumëfishtë ku mjafton vetëm njëri nga kushtet të plotësohet për t'u kthyer string-u (karta) në fjalë.

```

if (dyNrl=="34"|| dyNrl=="37"){
    return "American Express";
}
else if(dyNrl[0]=='4'){

```



```

    return "Visa";
}
else if(dyNrI=="51"||dyNrI=="52"||dyNrI=="53"||dyNrI=="54"||dyNrI=="55"){
    return "MasterCard";
}
else {
    return "Eshte karte tjeter";
}
}

```

### // 3) Funksioni "main"

*/\* Në funksionin kryesor "main" që shërben si "entry point" i programit do thërrasim funksionet që krijuam më sipër nëpërmjet argumenteve përkatëse. \*/*

```

int main()
{

    cout<<"1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.\n"
    <<"2) Shtyp 2 per informacionet rreth kartes.\n"
    <<"3) Shtyp 3 per te dale nga programi.\n\n";
    int zgjedhja;
    cin>>zgjedhja;           //pas deklarimit, fusim si input variablin zgjedhja

```

*/\**

*Për krijimin e një mini-menuje, brenda ciklit Do While do përdorim shprehjen Switch. Një nga rastet duhet të përputhet në bazë të vlerës së variablit zgjedhja që përdoruesi fut nga tastiera, përndryshe kalohet në default. Për çdo rast, përdorim fjalën kyçe break për të kaluar jashtë bllokut Switch ose **return 0**; për përfundimin e programit.*

*Pra, kodi i fundit (return 0;) do e mbyll menjëherë programin, ndërsa nëse shprehja brenda while(false), dilet nga cikli dhe mund të futet numri i kartës përkatësisht për case 1 ose case 2.*

*\*/*

```

do{
    switch(zgjedhja){
        case 1:
            break;

        case 2:
            cout<<"Kini parasysh qe karta juaj duhet te kete 10-19 shifra.\n";
            cout<<"P.sh nje karte me 16 shifra ka ne perberje: \n"
            <<"Shifra 1: nr i emetuesit\n"
            <<"Shifra 2-6: emri i bankes\n"
            <<"Shifra 7-12: nr i llogarise\n"
            <<"Shifra 13-16: kontrolli i shifrave\n";
            break;

        case 3:
            cout<<"Programi u mbyll!";
            return 0;

        default: cout<<"Perzgjidhet by default opsioni 1."<<endl;
    }
}

```

```
}  
} while (zgjedhja == 3);
```

```
string nrKarte;           //deklarimi i një string-u që do ruajë numrin e kartës
```

```
/*
```

Nisur nga fakti që një kartë duhet të jetë përmbajë nga 10-19 shifra, atëherë përmes ciklit Do While po përsërisim të paktën njëherë bllokun e kodit, e më pas do testohet kushti duke pyetur që "A është gjatësia e këtij string-u më e vogël se 10 (që përfshin dhe 0 dhe nr. negativë) ose më e madhe se 19? Për t'u përsëritur cikli, mjafton vetëm njëri prej këtyre kushteve të jetë True (p.sh në rast se fus një numër 20-shifror). Në të kundërt, në rast se të dyja kushtet janë False (kur futet numër 10-19-shifror), dilet nga cikli.

```
*/
```

```
do{
```

```
    cout<<"Ju lutem të fusni tani numrin e kartes (10-19): "<<endl; //përdorim objektin cout që si  
    output, afishohet/shkruhet mesazhi brenda thonjzave (shfaqet në monitor)
```

```
    cin>>nrKarte; //vlerën e variablit që përdoruesi shtyp nga keyboard, kopjoje te variab nrKarte
```

```
} while (nrKarte.size()<10 || nrKarte.size()>19);
```

// Kontrollojmë nëse numri i kartës është i vlefshëm, duke thirrur funksionet me argumentet përkatëse.

```
if (karteevlefshme(nrKarte)){
```

```
    cout<<"*****"<<endl;
```

```
    cout<<"KARTA ESHTË E VLEFSHME."<<endl;
```

```
    cout<<"LLOJI I KARTES: "<<setw(5)<<llojiKartes(nrKarte)<<endl;
```

```
    cout<<"*****"<<endl;
```

```
}
```

```
else{
```

```
    cout<<"KARTE E PAVLEFSHME."<<endl;
```

```
}
```

```
return 0; //mbyll ekzekutimin e programit (pa gabime)
```

```
}
```

#### 4. OUTPUT-et e programit (duke testuar vlera të ndryshme):

**Rasti 1)** Kur përdoruesi përzgjedh një numër negativ ose të ndryshëm nga 1-3, kalohet si "default" opsioni 1.

```
C:\Users\User\Desktop\DETYRE KURSI ALGORITMIKE DANIEL101\kod\MeKoment
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

-1
Perzgjidhet by default opsioni 1.
Ju lutem te fusni tani numrin e kartes (10-19):
```

```
C:\Users\User\Desktop\DETYRE KURSI ALGORITMIKE DANIEL101\kod\MeKoment
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

5
Perzgjidhet by default opsioni 1.
Ju lutem te fusni tani numrin e kartes (10-19):
```

**Rasti 2)** Kur përdoruesi përzgjedh numrin 1 të menisë, por fut vlera jo të sakta për kartën, derisa fut atë të duhurën.

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

1
Ju lutem te fusni tani numrin e kartes (10-19):
0
Ju lutem te fusni tani numrin e kartes (10-19):
485
Ju lutem te fusni tani numrin e kartes (10-19):
382684590347649775453644738
Ju lutem te fusni tani numrin e kartes (10-19):
4111111111111111
*****
KARTA ESHTË E VLEFSHME.
LLOJI I KARTES: Visa
*****
```

**Rasti 3)** Kur përdoruesi përzgjedh numrin 1 të menisë dhe fut numrin e tij të saktë të kartës, qoftë i vlefshëm ose i pavlefshëm. Në rast se është i vlefshëm, shfaqet gjithashtu dhe lloji i kartës.

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

1
Ju lutem te fusni tani numrin e kartes (10-19):
4012888888881881
*****
KARTA ESHTË E VLEFSHME.
LLOJI I KARTES: Visa
*****
```

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

1
Ju lutem te fusni tani numrin e kartes (10-19):
5555555555554444
*****
KARTA ESHTË E VLEFSHME.
LLOJI I KARTES: MasterCard
*****
```

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

1
Ju lutem te fusni tani numrin e kartes (10-19):
378282246310005
*****
KARTA ESHTË E VLEFSHME.
LLOJI I KARTES: American Express
*****
```

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

1
Ju lutem te fusni tani numrin e kartes (10-19):
6011111111111117
*****
KARTA ESHTË E VLEFSHME.
LLOJI I KARTES: Eshte karte tjeter
*****
```

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

1
Ju lutem te fusni tani numrin e kartes (10-19):
3012888888881881
KARTE E PAVLEFSHME.
```

**Rasti 4)** Përdoruesi zgjedh numrin 2 të menisë që t'i shfaqet një informacion paraprak lidhur me kartën.

```
C:\Users\user\Desktop\DETRE KURSI ALGJITIMIKE DANIEL TOI\KODIME\KODIME
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.
2) Shtyp 2 per informacionet rreth kartes.
3) Shtyp 3 per te dale nga programi.

2
Kini parasysh qe karta juaj duhet te kete 10-19 shifra.
P.sh nje karte me 16 shifra ka ne perberje:
Shifra 1: nr i autentisit
Shifra 2-6: emri i bankes
Shifra 7-12: nr i llogaritit
Shifra 13-16: kontrolli i shifrave
Ju lutem te fusni tani numrin e kartes (10-19):
```

### **Rasti 5)** Përdoruesi zgjedh të dalë nga programi.

```
1) Shtyp 1 per te kontrolluar vlefshmerine dhe llojin e kartes.  
2) Shtyp 2 per informacionet rreth kartes.  
3) Shtyp 3 per te dale nga programi.  
  
Programi u mbyll!
```

### **III. KONKLUZIONET**

- Arritëm të shfrytëzonim njohuritë e marra në funksion të zgjidhjes së problemit duke u mbështetur nga teoritë e mësuara. Kjo na nxiti të mendojmë në mënyrë kritike duke përdorur si numër karte tipin e të dhënës “string”, karakteret e të cilit na shërbyen si një mekanizëm për kontrollin e vlefshmërisë së kartës, si dhe për identifikimin e llojit të kartës bazuar mbi rregullat e kompanive të ndryshme.
- Aplikuam në rrafshin praktik përdorimin e cikleve, përzgjedhjeve, funksioneve, vektorëve etj duke u munduar që ta optimizonim kodin në sa më pak rreshta. Pra, këto elementë të programimit na erdhën në ndihmë si zgjidhje optimale për përmbysjen e string-ut, llogaritjen e veprimeve (si p.sh shuma e akumuluar) etj.
- Si synim i mëtejshëm, do dëshironim të implementonim në kod dhe në bllokskemë edhe mënyrën e dytë të algoritmit të Luhn-it për vlefshmërinë e kartave.
- Ndonëse është akoma herët, por si objektiv në projektet e tjera do synojmë ndërtimin e një sistemi më kompleks, duke zgjeruar menunë (opsionet) duke shtuar kështu mundësinë e ruajtjes së më shumë të dhënave (emrave, llogarive të ndryshme, datave), si edhe ndëraktivitetin me përdoruesin.

### **Bibliografia:**

Burimi (resource) i mëposhtëm na ndihmoi pjesërisht për zhvillimin e logjikës së vlefshmërisë së kartave duke u mbështetur te shpjegimi i hollësishëm i kartave të kreditit dhe algoritmit të Luhn-it:

- [A Comprehensive Guide to Validating and Formatting Credit Cards | by Kelvin Zhang | Hootsuite Engineering | Medium](#)
- <https://www.youtube.com/watch?v=3bBiQ1G7BOU>
- B. Chaudhuri, “Flowchart and Algorithm Basics”, Mercury Learning and Information, New Dehli, 2020.
- Behrouz A. & Richard F., “C++ Programming: An Object-Oriented Approach”, McGraw-Hill Education, New York, 2020.
- Ardiana T., “Leksioni V – Tipet e të dhënave & shprehjet në C++”, fq.12, 2022.

Vlerësimi i pedagogut:	Pikët:
I. Parashtrimi i tezës së esesë:	
II. Argumentimi:	
III. Përfundimi:	
<b>Totali</b>	