

INLÄMNINGSUPPGIFT 1 LP5

Roulett

IT120G Grundläggande programmering med c++
G1N

Anton Karlsson (19940916-7534)
C2oantka

2020-11-25

Innehåll

Introduktion	1
Problembeskrivning.....	1
Satsning utav pengar.....	1
Få reda på om ”bollen” landade på färgen svart eller röd.....	1
Nuvarande saldo	2
Att spela igen.....	2
När saldot tar slut.....	2
För stor satsning.....	2
Redovisning utav totala vinsten	2
Antaganden och krav.....	2
Lösningsdesign	3
Planering med hjälp utav flödesdiagram	3
Starta spelet	3
Satsningskrav.....	3
Saldokrav	3
Välja nummer eller färg.....	3
Räkna ut färg på genererade numret.....	3
Spela på nummer	3
Totalvinstmeddelande.....	4
Spela igen	4
Diskussion.....	6
Källkod	6

Introduktion

I denna rapport kommer det beskrivas hur man kan utveckla ett program för att kunna spela roulett, samt hur lösningen ser ut. För att kunna lösa uppgiften krävdes det att identifiera en god uppbyggnad med hjälp utav ett flödesdiagram. Med grundläggande kunskap om datatyper och funktioner så går det att bygga flödesdiagrammet runt de kunskaperna. Flödesdiagrammet ligger till grunden för kodens uppbyggnad.

Uppgiften var följande: Spelaren får en förfrågan om hen vill spela eller inte, sedan ska spelaren få välja en insats på 100, 300 eller 500kr förutsatt att spelaren svarat ja på spelförfrågan. Sedan ska spelaren få möjligheten att välja mellan att satsa på en färg (svart och röd) eller nummer. Sedan ska det slumpmässigt genererade numret att redovisas, samt uttalande om vinst eller förlust ska synas. Därefter ska även vinsten eller förlusten som hänt redovisas, samt den totala vinsten eller förlusten.

Efter uttalandet om totala vinsten/förlusten kommer en ny förfrågan om spelaren vill spela igen, om hen svarar ja kommer spelet att köras en gång till, om svaret är nej kommer spelet att avslutas. Programmet kan komma att avbrytas om saldot inte är lika med eller högre än insatsen, samt om saldot når noll kommer programmet att stoppas.

Problembeskrivning

Satsning utav pengar

Ett av problemen var att enbart acceptera insatserna 100, 300 eller 500kr. Alltså ska en insats på 400kr till exempel INTE gå att mata in, utan då ska spelaren meddelas att ett felaktigt värde har matats in. I sig är det inget problem men kravlistan säger tydligt att endast dessa värden ska kunna satsas.

Få reda på om "bollen" landade på färgen svart eller röd

Ett annat problem som är lite mer komplicerat är att räkna ut om den virtuella bollen landat på ett jämnt eller ojämnt nummer utan att manuellt lägga in alla nummer som är svarta och röda. Det är bättre att lösa det med en uträkning som kan skalas upp. Om till exempel spelet skulle vara mellan 1–1000 istället. Då blir det väldigt många nummer att sitta och skriva in i varje färg. Detta kommer diskuteras vidare under lösningsdesign. Detta är ett problem för att vi måste kunna räkna ut om bollen landat på röd eller svart under spelets omgång om spelaren valt att spela på en färg.

Nuvarande saldo

Det är viktigt att hålla koll på saldot i spelet, det är ett problem som ej får glömmas. För att spelaren inte ska kunna spela med till exempel 0kr på saldot, samt att vinsterna ska läggas på. Även insatserna måste dras från saldot. Detta är ett problem för att spelaren måste kunna hålla koll på sitt saldo under spelets gång för att veta hur mycket hen har kvar och kan spela för.

Att spela igen

När första rundan av spelet är avklarad så måste spelaren få spela igen om hen önskar så, då behöver man loopa spelet för att gå om igen. Då måste även saldot minnas för att inte återställas till "startsaldot" igen, så att vinsterna eller eventuella förluster räknas med i den nya spelomgången. Spelaren ska även få möjligheten att avsluta spelet efter spelomgången. Problemet här är spelet måste kunna spelas igen och inte avslutas efter en omgång precis som kravlistan beskriver. Lösningen för detta diskuteras under lösningsdesign.

När saldot tar slut

Om pengarna tar slut, alltså att saldot = 0 så måste spelet avbrytas. Eftersom inga insatser kan göras då de alla överstiger nuvarande saldo. Därför om spelaren vill spela igen men saldot är 0 så ska spelet avbrytas även om spelaren inte vill sluta spela. Problemet här är att spelaren inte ska kunna fortsätta spela utan pengar.

För stor satsning

Självklart ska man inte kunna spela för mer pengar än vad som finns på kontot (saldot). Därför måste det finnas ett krav att saldot är större än insatsen. Det finns ingen kreditering i spelet så därför ska inte en insats som är större än saldot vara möjligt. Problemet här är att se till att spelare inte spelar för mer pengar än vad hen har.

Redovisning utav totala Ovinsten

När man har spelat några omgångar och förhoppningsvis vunnit så ska man kunna se den totala vinsten som ackumulerats under omgångarna. Vilket kan bli ett problem om man inte sparar datan rätt under omgångarna. Problemet är att bara redovisa totala vinsten om saldot överstiger startsaldot.

Antaganden och krav

De första antaganden som kom att tänkas på var att kreditering ej kommer att tillämpas, så att insatsen ska inte kunna överstiga saldot samt att spelaren ska ej kunna fortsätta spelet när saldot nått 0. Sedan behöver det bestämmas vilka tal färgerna ska representera, i detta fall kommer svart att representera jämna nummer och röd ojämna nummer. Eftersom spelaren ska kunna spela flera gånger och samt få avsluta spelet när hen önskar så har antagandet tagits att spelaren bör frågas efter varje spelad omgång om hen vill spela igen. Samt innan spelet börjar så bör någon form utav hälsning ske samt förklaring vad som kommer att hända närmast, detta var ett dolt antagande men ändå en självklarhet. Då ett spel utan instruktioner blir väldigt svårt att spela korrekt, samt hur vinstmultiplikatorn ser ut. Därför har ett meddelande lagts till som skriver ut tydligt att rätt nummer ger 10x insatsen och rätt färg ger 2x insatsen, för att spelaren ska kunna ta med det i tankebanan när satsningen ska läggas.

Lösningsdesign

Planering med hjälp utav flödesdiagram

För att lösa uppgiften tänktes en stegvis plan ut, med hjälp utav flödesdiagrammet fick planen ett tydligt visuellt tillvägagångssätt.

Starta spelet

Första steget var att visa ett välkomstmeddelande som klargör vad för typ av program det är och vad som skall komma ske. Sedan blev nästa steg att fråga spelaren om hen vill spela, för att få ett enkelt sätt att avsluta programmet eller att fullfölja spelet (se figur 1). Därefter går programmet vidare till att fråga om storleken på insatsen med ett meddelande, sedan tas input från spelaren på önskad storlek (100,300 eller 500kr).

Satsningskrav

För att förhindra att andra nummer skrivs in och godkänns så har if satser använts så som `if(insats==100 || insats==300 || insats==500)`, om ett annat värde skrivs in går else satsen igång och skriver ut felaktigt värde. Spelaren får då välja om hen vill spela igen vid misslyckat val utav insats för att låta programmet fortgå och inte abrupt stängas ner vid fel inmatning.

Saldokrav

Nästa problem att lösa är att insatsen inte får vara större än tillgängligt saldo, det kan enkelt lösas med att utföra en koll på de två variablerna och se till att `saldo>insats`, även när `saldo==0` så kommer spelet att avslutas då pengarna är slut. Då när insatsen har lagts och godkänts så kommer nästa problem, nu ska spelaren kunna spela på nummer eller färg.

Välja nummer eller färg

För att lösa detta användes en fråga där spelaren får välja med input igen om hen vill satsa på en färg eller nummer, vi börjar med att kolla över satsning på färg. I denna lösning har datatypen `bool` använts för att senare jämföra vald färg med färg bollen landat på, så om spelaren väljer svart kommer variabeln "farg" sättas till 's'.

Räkna ut färg på genererade numret

I detta roulettspel är svart alla de jämna talen, för att räkna ut om svart eller rött vunnit används en funktion som tar det genererade numret och använder sig utav modulus på talet med två för att enkelt kunna räkna ut om modulus är 0 är det då jämnt, om modulus inte blir 0 är det ojämnt vid delning utav två. Här är ett exempel på modulus funktionen `if(number % 2 == 0)` så sätts variabeln `black` till `true`, else `false`. Sedan jämförs den valda färgen med `bool` variabeln `black`, så stämmer färgvalet svart med att `black = true` så sker vinst.

Spela på nummer

En liknande princip sker vid val av nummer fast den är mycket enklare, då behövs bara det valda numret jämföras med det genererade numret, om de är lika med varandra så utdelas vinst, else förlust. Vid vinst ska vinstsumman skrivas ut samt vid förlust ska den förlorade insatsen redovisas, vinstsumman blir då insatsen multiplicerat med två om spelaren spelat på färg respektive multiplicerat med tio om spelaren vann med valt nummer.

Totalvinstmeddelande

Vid flera spelade omgångar ska totalvinsten redovisas, då har det problemet lösts med att använda sig utav nuvarande saldot minus startsaldot, för att redovisa differensen mellan de två, för att inte visa vid förluster har if sats använts där det bara redovisas om saldot är över 1000, alltså startsaldot. Sedan lades en del till ett output som redovisar hur mycket minus spelaren ligger vid förluster, som dock ej var ett krav i uppgiften men som lades till ändå.

Spela igen

Efter avslutat spel så ställs frågan om spelaren vill spela igen, då tas input på 'svar' variabeln igen för att sätta igång eller avbryta while-loopen. Spelaren kan då alltså köra spelet (spel loopen) tills hen har 0 saldo eller väljer att svara nej på frågan efter avslutad spelomgång. Då kommer spelloopen att avslutas och ett hejdå meddelande går ut till spelaren, och värdet 0 returneras. Se figur 1 för att enklare förstå programmet och dess steg nedan.

Diskussion

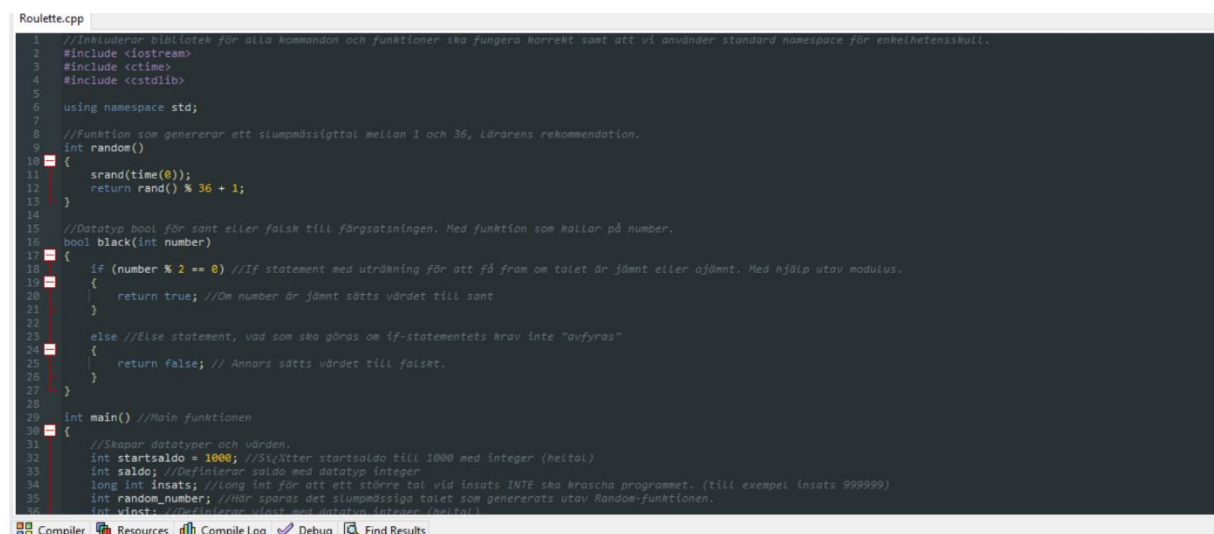
Lösningen som visas i källkoden har vissa svagheter som hade gjorts annorlunda om programmet skrivits igen. Till exempel, hade det varit bättre att bygga upp programmet med fler funktioner för att hålla isär allting och lättare strukturera upp flödesdiagrammet med en ruta för varje funktion. En svaghet är att input för svar tas i char, så det ser inte så proffsigt ut att bara ta in en bokstav vid svar, men gjorde för enkelhetens skull då det är enklare att alltid skriva rätt för spelaren.

Andra nackdelar är att programmet inte är optimerat för felaktiga värden, att ge en input vid satsning som är ett annat tal än 100,300 och 500 fungerar bra men skulle spelare börja svara andra svar än färg, nummer, ja och nej så kommer programmet inte fungera korrekt. Men följer man instruktionerna som ges vid körning utav programmet så fungerar det enligt kraven som är satta för uppgiften.

Styrkor med lösningen är enbart dess funktion, det fungerar och är spelbart, men som nämnt tidigare så kan en betydligt bättre version göras nu i efterhand med de nya kunskaper som lärts ut under kursens gång.

Källkod

Nedan ses skärmbilder med källkoden samt kommentarer, för enklare läsning rekommenderas cpp filen istället.



```
Roulette.cpp
1 //Inkluderar bibliotek för alla kommandon och funktioner ska fungera korrekt samt att vi använder standard namespace för enkelhetens skull.
2 #include <iostream>
3 #include <ctime>
4 #include <cstdlib>
5
6 using namespace std;
7
8 //Funktion som genererar ett slumpmässigt tal mellan 1 och 36, lärarens rekommendation.
9 int random()
10 {
11     srand(time(0));
12     return rand() % 36 + 1;
13 }
14
15 //Datatyp bool för sant eller falsk till försgatsningen. Med funktion som kallar på number.
16 bool black(int number)
17 {
18     if (number % 2 == 0) //If statement med uträkning för att få fram om tolet är jämnt eller ojämnt. Med hjälp utav modulus.
19     {
20         return true; //Om number är jämnt sätts värdet till sant
21     }
22
23     else //Else statement, vad som ska göras om if-statementets krav inte "uppfylls"
24     {
25         return false; // Annars sätts värdet till falskt.
26     }
27 }
28
29 int main() //Main funktionen
30 {
31     //Skapar datatyper och värden.
32     int startsaldo = 1000; //Sätter startsaldo till 1000 med integer (heltal)
33     int saldo; //Definierar saldo med datatyp integer
34     long int insats; //Long int för att ett större tal vid insats INTE ska krascha programmet. (till exempel insats 999999)
35     int random_number; //Här sparas det slumpmässiga tolet som genererats utav Random-funktionen.
36     int vinst; //Definierar vinst med datatyp integer (heltal)
```



```

Roulette.cpp
28
29 int main() //Main funktionen
30 {
31     //Skapar datatyper och värden.
32     int startsaldo = 1000; //Sätter startsaldo till 1000 med integer (hejtal)
33     int saldo; //Definierar saldo med datatyp integer
34     long int insats; //Long int för att ett större tal vid insats INTE ska krascha programmet. (till exempel insats 999999)
35     int random_number; //Här sparas det slumpmässiga talet som genererats utav Random-funktionen.
36     int vinst; //Definierar vinst med datatyp integer (hejtal)
37     char svar; //Definierar svar med datatyp char, för att lagra en bokstav som svarsalternativ.
38     saldo = startsaldo; //Sätter saldo till samma värde som startsaldo.
39     long int totalvinst; //Definierar totalvinst med long int ifall spelaren vinner mycket.
40
41     //Välkomstmeddelande och saldo information.
42     cout << "Hej och välkommen till att spela Roulette " << endl;
43     cout << "Ditt startsaldo är: " << startsaldo << "kr" << endl;
44
45     //Förfrågan om spelaren vill spela och tar emot input
46     cout << "Vill du spela? (J/N) " << endl;
47     cin >> svar;
48
49
50
51
52 //While funktion så att spelet bara fungerar medan spelaren svarat Ja och saldot är över värdet 0.
53 while (svar == 'J' || svar == 'j' and saldo > 0)
54 {
55     cout << "Vad kul att du vill spela, var snäll och ange storlek på din insats (100,300 eller 500kr) " << endl; //Meddelande om insatsstorlekar
56     cin >> insats; //Input insats (insatsstorlek)
57
58     if (insats > saldo) //If statement som går igång om insatsen är större än saldo
59     {
60         cout << "Du kan inte satsa mer än ditt tillgängliga saldo!" << endl; //Varningsmeddelande
61     }
62
63     else if (insats == 100 || insats == 300 || insats == 500) //Om insatsen stämmer med satta insatskraven
64     {
65         random_number = random(); //Random number sätts till värdet som random funktionen genererat.
66         cout << "Du har satsat: " << insats << "kr" << endl; //Output hur mycket spelaren har satsat
67         saldo = saldo - insats; //Saldo justeras efter insats
68         cout << "Ditt nuvarande saldo är: " << saldo << endl; //Output på saldo efter justering
69         cout << "Vill du välja ett nummer eller färg (N för nummer, F för färg)" << endl; //Fråga om spelare vill spela p färg eller nummer
70         cout << "Notera att rätt nummer ger 10x insatsen, rätt färg ger 2x insatsen" << endl; //Output om vinst multiplikatorn
71         char val; //Definierar val med datatypen char
72         cin >> val; //Input för om spelaren vill satsa på färg eller nummer, sparas som bokstav i val.
73
74         if (val == 'N' || val == 'n') //Om valet är N (Nummer)
75         {
76             cout << "Du får nu välja ett nummer mellan 1 - 36 " << endl; //Meddelande om att spelaren ska välja nummer
77             int nummer;
78             cin >> nummer; //Input anskat nummer spelaren vill satsa på
79             cout << "Du har valt: " << nummer << endl; //Output där spelaren ser vilket nummer hen har valt
80             cout << "Lycka till! \n" << endl; //Meddelande lycka till
81
82             if (random_number == nummer) //Om det slumpmässiga talet som genererats är samma som det valda numret (vinst)
83             {
84                 vinst = (insats*10); //Vinsten räknas ut med insats multiplicerat med 10
85                 saldo = saldo + vinst; //Saldo uppdateras med nuvarande saldo + vinsten.
86                 cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om vilket nummer som genererades
87             }
88         }
89     }
90 }

```

```

Roulette.cpp
51
52 //While funktion så att spelet bara fungerar medan spelaren svarat Ja och saldot är över värdet 0.
53 while (svar == 'J' || svar == 'j' and saldo > 0)
54 {
55     cout << "Vad kul att du vill spela, var snäll och ange storlek på din insats (100,300 eller 500kr) " << endl; //Meddelande om insatsstorlekar
56     cin >> insats; //Input insats (insatsstorlek)
57
58     if (insats > saldo) //If statement som går igång om insatsen är större än saldo
59     {
60         cout << "Du kan inte satsa mer än ditt tillgängliga saldo!" << endl; //Varningsmeddelande
61     }
62
63     else if (insats == 100 || insats == 300 || insats == 500) //Om insatsen stämmer med satta insatskraven
64     {
65         random_number = random(); //Random number sätts till värdet som random funktionen genererat.
66         cout << "Du har satsat: " << insats << "kr" << endl; //Output hur mycket spelaren har satsat
67         saldo = saldo - insats; //Saldo justeras efter insats
68         cout << "Ditt nuvarande saldo är: " << saldo << endl; //Output på saldo efter justering
69         cout << "Vill du välja ett nummer eller färg (N för nummer, F för färg)" << endl; //Fråga om spelare vill spela p färg eller nummer
70         cout << "Notera att rätt nummer ger 10x insatsen, rätt färg ger 2x insatsen" << endl; //Output om vinst multiplikatorn
71         char val; //Definierar val med datatypen char
72         cin >> val; //Input för om spelaren vill satsa på färg eller nummer, sparas som bokstav i val.
73
74         if (val == 'N' || val == 'n') //Om valet är N (Nummer)
75         {
76             cout << "Du får nu välja ett nummer mellan 1 - 36 " << endl; //Meddelande om att spelaren ska välja nummer
77             int nummer;
78             cin >> nummer; //Input anskat nummer spelaren vill satsa på
79             cout << "Du har valt: " << nummer << endl; //Output där spelaren ser vilket nummer hen har valt
80             cout << "Lycka till! \n" << endl; //Meddelande lycka till
81
82             if (random_number == nummer) //Om det slumpmässiga talet som genererats är samma som det valda numret (vinst)
83             {
84                 vinst = (insats*10); //Vinsten räknas ut med insats multiplicerat med 10
85                 saldo = saldo + vinst; //Saldo uppdateras med nuvarande saldo + vinsten.
86                 cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om vilket nummer som genererades
87             }
88         }
89     }
90 }

```

```

Roulette.cpp
111 cout << "Korrekt färg ger två gånger pengarna" << endl; //Meddelande om potentiella vinst multiplikationen
112 if (black(random_number) == false) //Om numret som genererats i random_number är ojämnt räknas det som rätt och vinst sker.
113 {
114     cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om vilket nummer som har genererats
115     vinst = (insats*2); //Uträkning utav vinst
116     saldo = saldo + vinst; //Uppdatering utav saldo
117     cout << "Grattis du vann! \n" << endl; //Vinstmeddelande med alert
118     cout << "Vinst: " << (vinst - insats) << "kr" << endl; //Meddelande om reella vinsten
119     cout << "Ditt nuvarande saldo är: " << saldo << "kr" << endl; //Meddelande om nuvarande saldo
120 }
121 else //Om det inte är jämnt nummer händer detta (förlust)
122 {
123     cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om genererat nummer
124     cout << "Oturl! Du förlorade!" << endl; //Förlust meddelande
125     cout << "Förlust: " << insats << "kr" << endl; //Meddelande om insatsen som förlorats
126     cout << "Ditt nuvarande saldo är: " << saldo << "kr" << endl; //Meddelande om nuvarande saldo
127 }
128 }
129 else if (farg == 'S' || farg == 's') //Om valt färg är S (svart) sker detta
130 {
131     cout << "Du valde färgen Svart" << endl; //Meddelande om vald färg
132     cout << "Korrekt färg ger två gånger pengarna" << endl; //Meddelande om vinst multiplikatorn
133     if (black(random_number) == true) //Om talet är jämnt (true) och räknas som svart sker detta
134     {
135         cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om numret som genererats
136         vinst = (insats*2); //Uträkning för vinsten
137         saldo = saldo + vinst; //Uppdatering utav saldo efter vinst
138         cout << "Grattis du vann! \n" << endl; //Vinst meddelande
139         cout << "Vinst: " << (vinst - insats) << "kr" << endl; //Meddelande om vinsten som vunnits
140         cout << "Ditt nuvarande saldo är: " << saldo << "kr" << endl; //Meddelande om nuvarande saldo efter vinst
141     }
142     else //Vad som händer om talet inte är jämnt alltså false sker detta
143     {
144         cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om numret som genererats
145         cout << "Oturl! Du förlorade!" << endl; //Meddelande om förlust
146         cout << "Förlust: " << insats << "kr" << endl; //Meddelande om insatsen som förlorats
147     }
148 }

```

```

Roulette.cpp
111 cout << "Korrekt färg ger två gånger pengarna" << endl; //Meddelande om potentiella vinst multiplikationen
112 if (black(random_number) == false) //Om numret som genererats i random_number är ojämnt räknas det som rött och vinst sker.
113 {
114     cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om vilket nummer som har genererats
115     vinst = (insats*2); //Uträkning utav vinst
116     saldo = saldo + vinst; //Uppdatering utav saldo
117     cout << "Grattis du vann! \n " << endl; //Vinstmeddelande med alert
118     cout << "Vinst: " << (vinst - insats) << "kr" << endl; //Meddelande om reella vinsten
119     cout << "Ditt nuvarande saldo är: " << saldo << "kr" << endl; //Meddelande om nuvarande saldo
120 }
121 else //Om det inte är jämnt nummer händer detta (förlust)
122 {
123     cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om genererat nummer
124     cout << "Otur! Du förlorade! " << endl; //Förlust meddelande
125     cout << "Förlust: " << insats << "kr" << endl; //Meddelande om insatsen som förlorats
126     cout << "Ditt nuvarande saldo är: " << saldo << "kr" << endl; //Meddelande om nuvarande saldo
127 }
128 }
129 else if (färg == 'S' || färg == 's') //Om valt färg är S (svart) sker detta
130 {
131     cout << "Du valde färgen Svart" << endl; //Meddelande om vald färg
132     cout << "Korrekt färg ger två gånger pengarna" << endl; //Meddelande om vinst multiplikation
133     if (black(random_number) == true) //Om tolet är jämnt (true) och räknas som svart sker detta
134     {
135         cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om numret som genererats
136         vinst = (insats*2); //Uträkning för vinsten
137         saldo = saldo + vinst; //Uppdatering utav saldo efter vinst
138         cout << "Grattis du vann! \n " << endl; //Vinst meddelande
139         cout << "Vinst: " << (vinst - insats) << "kr" << endl; //Meddelande om vinsten som vunnits
140         cout << "Ditt nuvarande saldo är: " << saldo << "kr" << endl; //Meddelande om nuvarande saldo efter vinst
141     }
142     else //Vad som händer om tolet inte är jämnt alltså false sker detta
143     {
144         cout << "Bollen landade på nummer: " << random_number << endl; //Meddelande om numret som genererats
145         cout << "Otur! Du förlorade! " << endl; //Meddelande om förlust
146         cout << "Förlust: " << insats << "kr" << endl; //Meddelande om insatsen som förlorats
147     }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }

```

```

Roulette.cpp
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }

```

```

Roulette.cpp
155 {
156     cout << "Felaktigt värde" << endl;
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }

```