Project 2

Title

Roulette

Modified version of a roulette game

Course

CIS-5CSC5

Due date

6/6/2016

Author

Edgar Salas

Introduction

Title : Roulette

This is modified version of regular game of roulette
The program prints out a roulette board with the columns and street bets labeled
The user enters the amount of each chip denomination they want to bet and what kind of bet they want to place
The program then uses randomized search function to simulate a roulette wheel spinning
Next it determines whether or not the bet was correct or not
It then calculates the resulting balance of gambling money
The user can then decides whether or not they want to continue playing
And lastly outputs the statistics of the game to a file

Summary

Project size : about 580 lines
The number of variables: about 24

This project includes all if not most of the things we have covered in class
I have improved on the previous version of this game by adding functions and adding a 2d array to hold the values of the board rather than just having a for loop printing out a incremented int value.

The hardest part of programming this game was making sure all the input was valid and keeping an correct betting total
As you can see in past versions I tried multiple ways of making sure that the input was valid and didn't crash the game if you entered anything besides a valid bet
I ended up using the isdigit function to check If the input was valid which let simply terminate the game rather than have it go through an endless loop

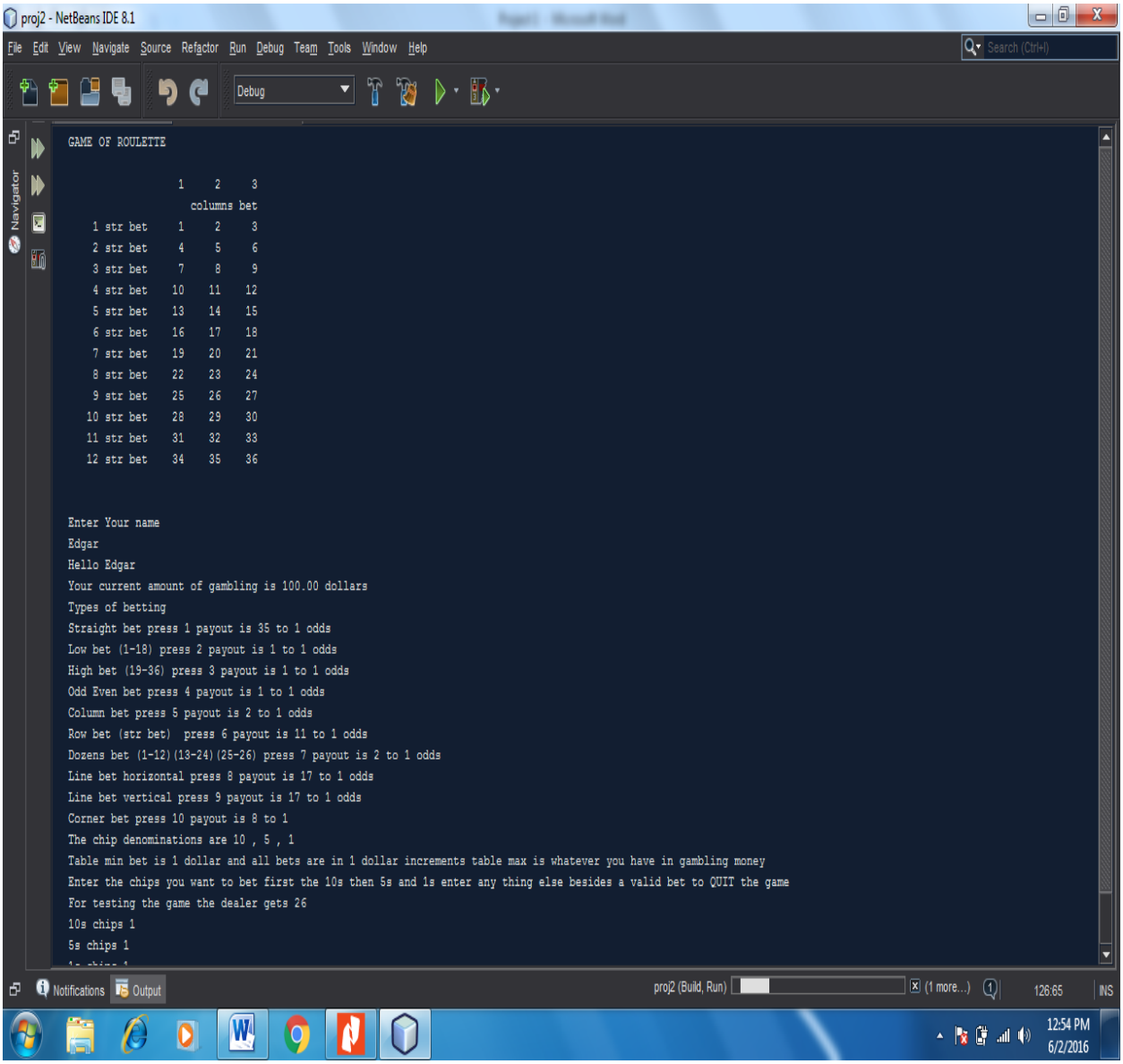
Pseudocode

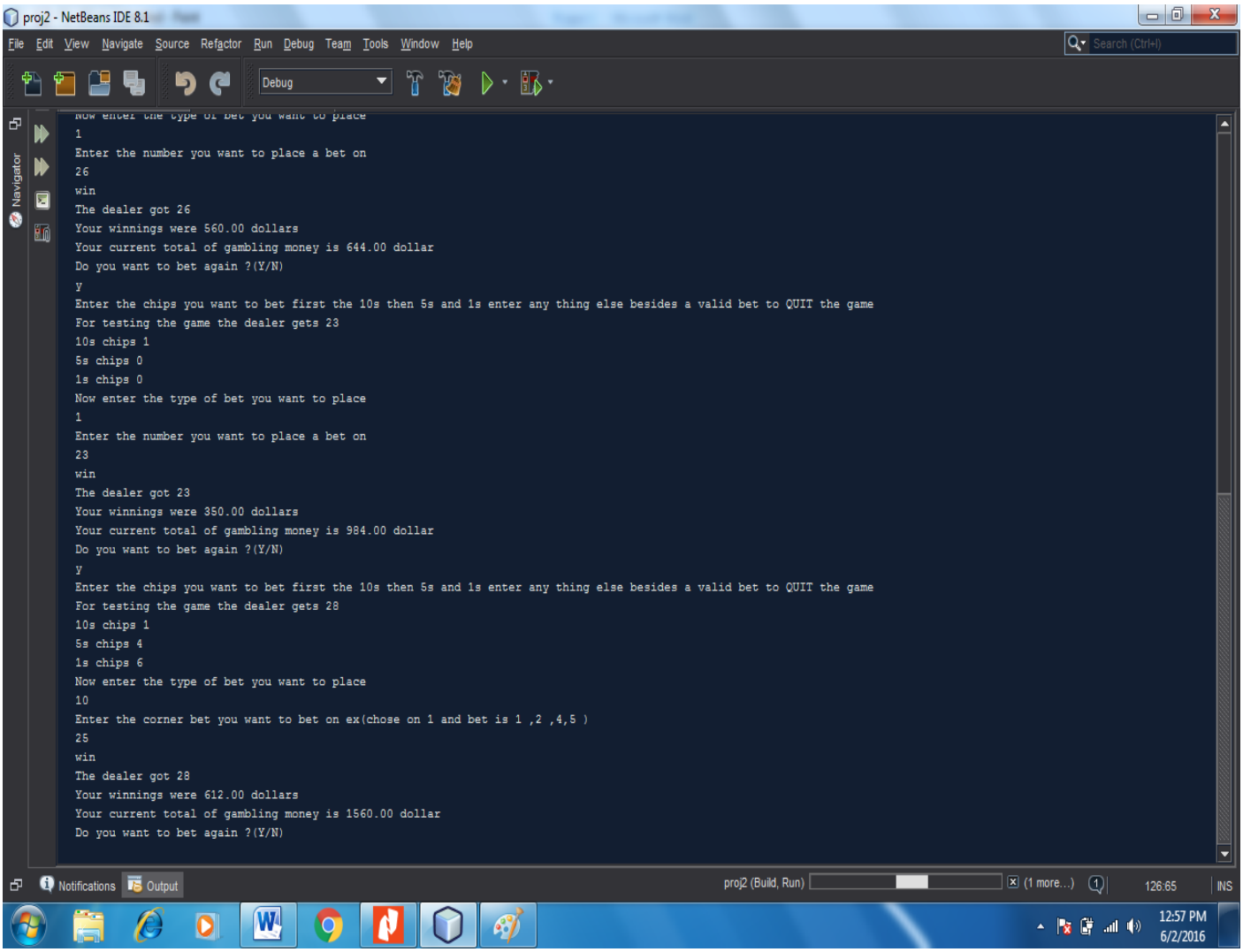
Declare all variables
Print out game title and board using for loops
Have user enter their name
Display rules for game and start do while loop for the game
Set spin to random numbers and ask for their betting chips
If input is valid then calculate the bet
Else end terminate game

Ask user what they want to bet on
If input is valid then check if the bet is correct
Else end terminate game
Use switch statement and nested if statements and nested switch statements to check if the bet is correct
Case 1 straight bet 35 to 1 odds
Case 2 low bet 1 to 1 odds
Case 3 high bet 1 to 1 odds
Case 4 odd even bet 1 to 1 odds
Case 5 column bet 2 to 1 odds
Case 6 row bet 11 to 1 odds
Case 7 dozens bet 2 to 1 odds
Default tell user they didn't pick a bet and terminate the game
Tell user whether they won or lost the bet and calculate their total Of gambling money
Ask user if they want to play again if yes then then repeat the betting process if they don't want to play again exit the game loop

Print the statistics for the game to a file and terminate the game

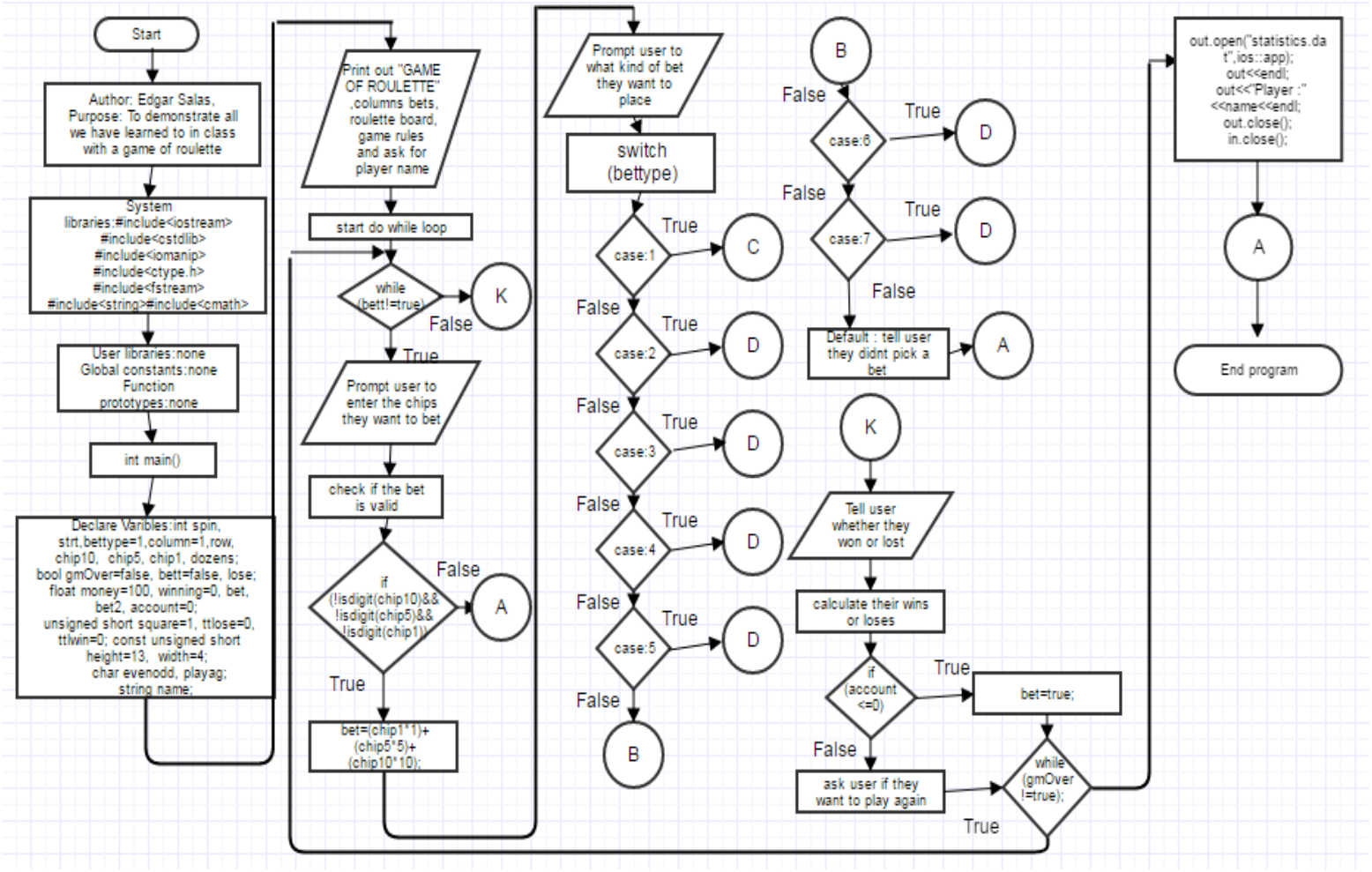
Screenshots

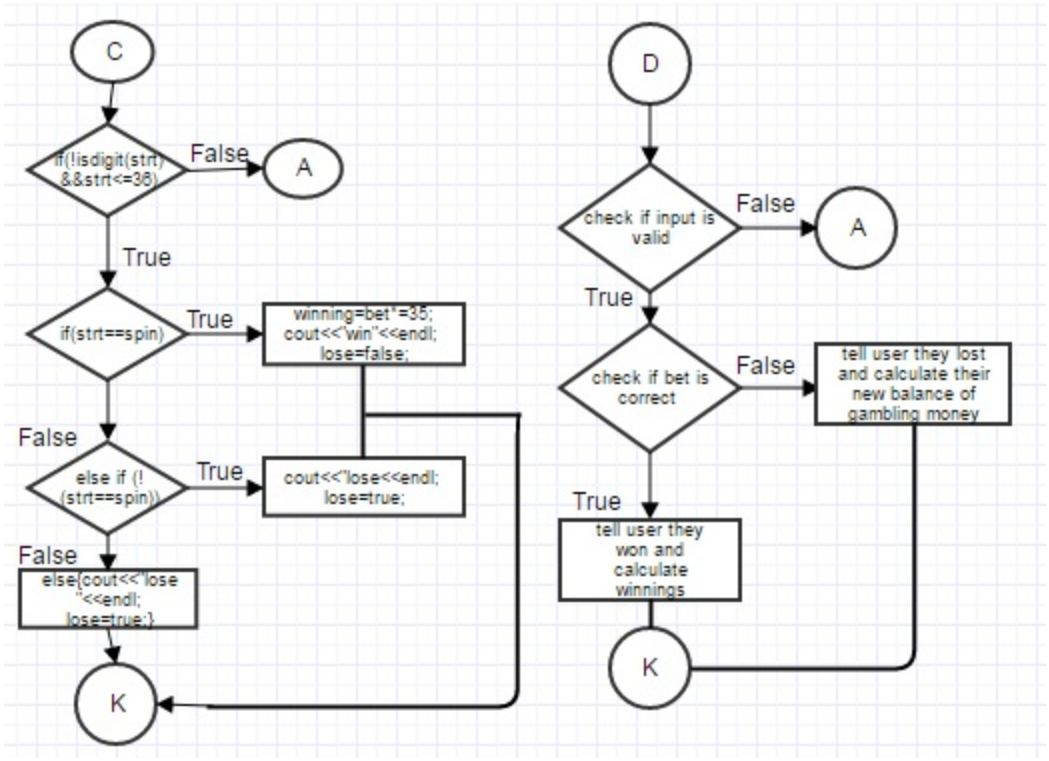



proj2 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Search (Ctrl+I)
Debug
Navigator
Now enter the type of bet you want to place
1
Enter the number you want to place a bet on
26
win
The dealer got 26
Your winnings were 560.00 dollars
Your current total of gambling money is 644.00 dollar
Do you want to bet again ?(Y/N)
y
Enter the chips you want to bet first the 10s then 5s and 1s enter any thing else besides a valid bet to QUIT the game
For testing the game the dealer gets 23
10s chips 1
5s chips 0
1s chips 0
Now enter the type of bet you want to place
1
Enter the number you want to place a bet on
23
win
The dealer got 23
Your winnings were 350.00 dollars
Your current total of gambling money is 984.00 dollar
Do you want to bet again ?(Y/N)
y
Enter the chips you want to bet first the 10s then 5s and 1s enter any thing else besides a valid bet to QUIT the game
For testing the game the dealer gets 28
10s chips 1
5s chips 4
1s chips 6
Now enter the type of bet you want to place
10
Enter the corner bet you want to bet on ex(chose on 1 and bet is 1 ,2 ,4,5)
25
win
The dealer got 28
Your winnings were 612.00 dollars
Your current total of gambling money is 1560.00 dollar
Do you want to bet again ?(Y/N)
Notifications
Output
proj2 (Build, Run)
(1 more...)
126:65
INS
12:57 PM
6/2/2016

C++ constructs

Chapter (Gaddis)	Keywords and syntax (v,x,y)=variables	location (line)
2.2	cout<<endl;	(63)cout<<GAME OF ROULETTE<<endl;
2.3	#include	(9-15)#include<iostream>
2.4	int	(23-31)int spin,etc;
2.5	true/false	(33)gmOver=false;
2.6	short/unsigned short	(42-46)unsigned short square, etc;
2.7	char	(52)char evenod,play again;
2.8	string	(54)string name;
2.9	float	(37-41)float money=100;
2,10	bool	(33-35)bool gmOver=false, etc;
2.14	(+-=*%)	(363) money+=bet;
2.16	const	(47-49)const unsigned short height=13;
3.1	cin>>	(100) cin>>chip10;
3.2	x=v+y;	(104)bet=(chip1*1)+(chip5*5)+(chip10*10);
3.6	v+=v v-=v v*=v	(127)winning=bet*=35;
3.7	cout<<setw()	(72)cout<<setw(6)<<square++;
3.8	getline(cin, v)	(80) getline(cin, name);
4.1	(&& ||)	(111)bettype<!8||bettype>8;
4.2	if(v<x){expression}	(70) if (i==0)cout<<setw(10)<<t;
4.4	if(v<x){expression} else{}	(70-73) if (i==0)cout<<setw(10)<<t; else cout<<setw(6)<<square++;
4.5	if(v<x){ if(y>x) {expression} }	(68-70)if (t==1) { if (i==0) {}}
4.6	if(x>y)	(136-139)if (spin>=19)else if (spin>=1&&spin<=18
4.7	if(v||v){expression}	(224)if (spin==1||spin==2||spin==3)
4.8	if(v<=v&&v>=v){expression}	(222)if(!isdigit(row)&&row>=12)
4,10	if(){ if() {expression} }else{expression}	(70)if (t==1) if (i==0)cout<<setw(10)<<t; else cout<<setw(6)<<square++;
4.13	(v)? Expression :expression	(363) (lose!=true)? (money+=bet, account=money-=bet2, ttlwin++) :(account=money-=bet2, ttlose++);
4.14	switch() case1:expression	(117)switch(bettype)
5.1	v++, v--	(72)square++;
5.2	while(x>y){expression}	(99)while(bett!=true){}
5.3	while(x>y&&x<y){expression}	(111)bettype<!8||bettype>8
5.4	while(v<=v||v>=v){v++}	(96&&363&&364)while(gmOver!=true){ttlose++, ttlwin}
5.5	do{expression}while(v<=v&&v>=v);	(108-379)do{}while(gmOver!=true)
5.6	for(v=1;v<v;v++){expression}	(66) for(short i=1;i!=height;i++)
5.7	for(v=1;v<v;v++){v+=v}	(108&&113)do{} while(bettype<!8||bettype>8);
5.8	while(x!=y){expression}	(113) while(bettype<!8||bettype>8);
5,10	while(v<x){for(v=1;v<v;v++){expression}	(66-67) for(short i=1;i!=height;i++){ for(short t=1;t!=width;t++){
5.11	ofstream out; out.open("v.dat")	(380-388) out.open("statistics.dat",ios::app); out<<endl; out<<"Player : "<<name<<endl;
6.3	int function(int x, int y)	(18-26) int wheel();
6.4	int function(x, y)	(535) swap(board [k][m] ,board [i][j]);
6.8	return x	(580) return spin
6.13	int function(int &x, int &y)	int swap (int & , int &);
7.1	array[]={x,y,z}	(482-495) board[i][j]=num++;
7.2	array[1]=array2[2]	(515)board[i][j]=board[column][row];
7.4	const int x=100; int y[x];	(78&&80) const int limit=50; int num=1;
7.9	array [][]={x,y,z}	(80) int board [limit][limit];
8.1	linear or binary	(507-521) int randarray(int board[][50])
8.3	sorts	(522-541)int sortarray(int board[][50])
TOTAL		


Start
Author: Edgar Salas, Purpose: To demonstrate all we have learned to in class with a game of roulette
System libraries:#include<iostream> #include<cstdlib> #include<iomanip> #include<ctype.h> #include<fstream> #include<string>#include<cmath>
User libraries:none Global constants:none Function prototypes:none
int main()
Declare Variables:int spin, strt,bettype=1,column=1,row, chip10, chip5, chip1, dozens; bool gmOver=false, bett=false, lose; float money=100, winning=0, bet, bet2, account=0; unsigned short square=1, ttlose=0, ttlwin=0; const unsigned short height=13, width=4; char evenodd, playag; string name;
Print out "GAME OF ROULETTE" ,columns bets, roulette board, game rules and ask for player name
start do while loop
while (bett!=true)
False
True
K
Prompt user to enter the chips they want to bet
check if the bet is valid
if (!isdigit(chip10)&& !isdigit(chip5)&& !isdigit(chip1))
False
A
True
bet=(chip1*1)+ (chip5*5)+ (chip10*10);
Prompt user to what kind of bet they want to place
switch (bettype)
case:1
case:2
case:3
case:4
case:5
True
False
C
D
B
case:6
case:7
Default : tell user they didnt pick a bet
K
Tell user whether they won or lost
calculate their wins or loses
if (account <=0)
bet=true;
ask user if they want to play again
while (gmOver !=true);
out.open("statistics.dat",ios::app); out<<endl; out<<"Player :" <<name<<endl; out.close(); in.close();
A
End program


C
if(!isdigit(strt) &&strt<=36)
False
A
True
if(strt==spin)
True
winning=bet*35; cout<<"win"<<endl; lose=false;
False
else if (! (strt==spin))
True
cout<<"lose<<endl; lose=true;
False
else{cout<<"lose "<<endl; lose=true;}
K
D
check if input is valid
False
A
True
check if bet is correct
False
tell user they lost and calculate their new balance of gambling money
True
tell user they won and calculate winnings
K