

Let's try writing a function!

Let's write a general algorithm for the game of blackjack.

keeping it simple, we will have one user play against the computer...

Welcome the user and explain the game

Deal two cards to the player

Allow the player to ask for another card ("hit") until they are satisfied

Player shows their hand

Dealer shows their hand

Do they want to play again or end the game?

PLAY A GAME

- Rules

• Playing the game
until winner or...

• Again?

- Termination.

main



"Rule"

Welcome

```
#include <iostream>
using namespace std;
// Program written by Karla Fant for CS161
// This program will allow one user to play a simple game of
// Blackjack against the computer
void welcome (); //This is where the rules will be described
int main()
{
    welcome(); //calling the function to display the rules
    cin.get();
    return 0;
}
```

//Explaining the rules of the game

```
void welcome()
```

```
{
```

'y'

```
char rules='y'; //do you want to see the rules?
```

```
cout <<"Welcome to the game of Black Jack" <<endl;
```

```
cout <<"The game will start soon, would you like"
```

<<" to hear about the rules? y/n ";

```
cin >> rules; cin.get();
```

rules = tolower(rules);

```
if (rules == 'y') //let's display the rules
```

```
{ cout <<endl <<endl
```

```
<<"*****"
```

```
<<endl <<endl;
```

```
cout <<"The goal of the game is to get as close to 21 as"
```

```
<<endl
```

```
<<"possible and get a higher number than the computer."
<<endl <<endl
<<"You will be given 2 cards."
<<endl <<"You can ask"
<<" to be hit with additional cards until you are"
<<" happy or over 21. "
<<endl <<endl;
}
cout <<"Let's begin!" <<endl <<endl;
}
```

```
#include <iostream>
using namespace std;
```

```
void welcome();
```

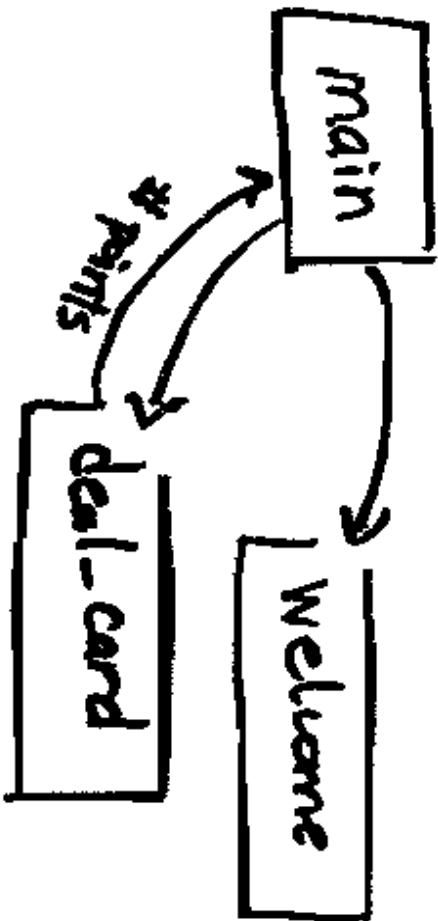
```
int deal_card();
```



```
int main() int user-points;
```

```
{  
    ...  
    user-points = deal_card();
```






```
void welcome (); //This is where the rules will be described
int deal_card(); //get a card and find out the points...
int main()
{
    int user_points; //to hold the number of points the player has
    srand(time(0)); SEEDING
    welcome(); //calling the function to display the rules
    user_points = deal_card(); //Deal a card

    cout <<"You have " <<user_points <<" points" <<endl;
    cin.get();
    return 0;
}
```

Beginning of matrix
strand (this (phi));

↓
Seed random # generator

≡
Number = rand () ;
variable

%

Mod Operator
Remainder

14

~~13~~

Number

0-13

Remainder

~~0-12~~

OPD

$$\begin{array}{r} 2 \overline{) 357} \\ \underline{357} \\ 0 \end{array}$$

Quotient
① Remainder

$$\begin{array}{r} 2 \overline{) 3} \\ \underline{3} \\ 0 \end{array}$$

Remainder 1

$$\begin{array}{r} 2 \overline{) 356} \\ \underline{356} \\ 0 \end{array}$$

Quotient
∅ Remainder

$$\begin{array}{r} 2 \overline{) 4} \\ \underline{4} \\ 0 \end{array}$$

R ∅



```

//Deal a card - one of 13 cards, 1 is Ace, 11 is Jack, 12 Queen, 13 King
int deal_card() // Correction to Slide 26
{
    int card = 0; //Find the numeric value of a card
    int points = 0; 0-12 + 1 } 1-13
    card = rand() % 13 + 1; //Make sure the card is within range 0-13
    if (1 == card) //ACE!!
    {
        do //find out the value they want to apply to the ace
        {
            cout <<"You have: an Ace " <<endl
            <<"Do you want to count it as"
            <<" a 1 or 11? ";
            cin >> points; cin.get();
        } while (points != 1 && points != 11); //it has to be correct!
    }
}

```

```
} else if (11 == card)//Jack!  
{  
    cout <<"You have: a Jack worth 10 points" <<endl;  
    points = 10;  
}  
} else if (12 == card) //Queen!  
{  
    cout <<"You have: a Queen worth 10 points" <<endl;  
    points = 10;  
} else if (13 == card) //King!  
{  
    cout <<"You have: a King worth 10 points" <<endl;  
    points = 10;  
}
```

```
} else  
{
```

```
    cout <<"You were given a face card: " <<card <<endl;
```

```
    points = card;
```

```
}
```

```
    return points;
```

```
}
```

```
//Deal a card - one of 14 cards, 0 is Ace, 11 is Jack, 12 Queen, 13 King
```

```
int deal_card()
```

```
{
```

```
    int card = 0;    //Find the numeric value of a card
```

```
    int points = 0;
```

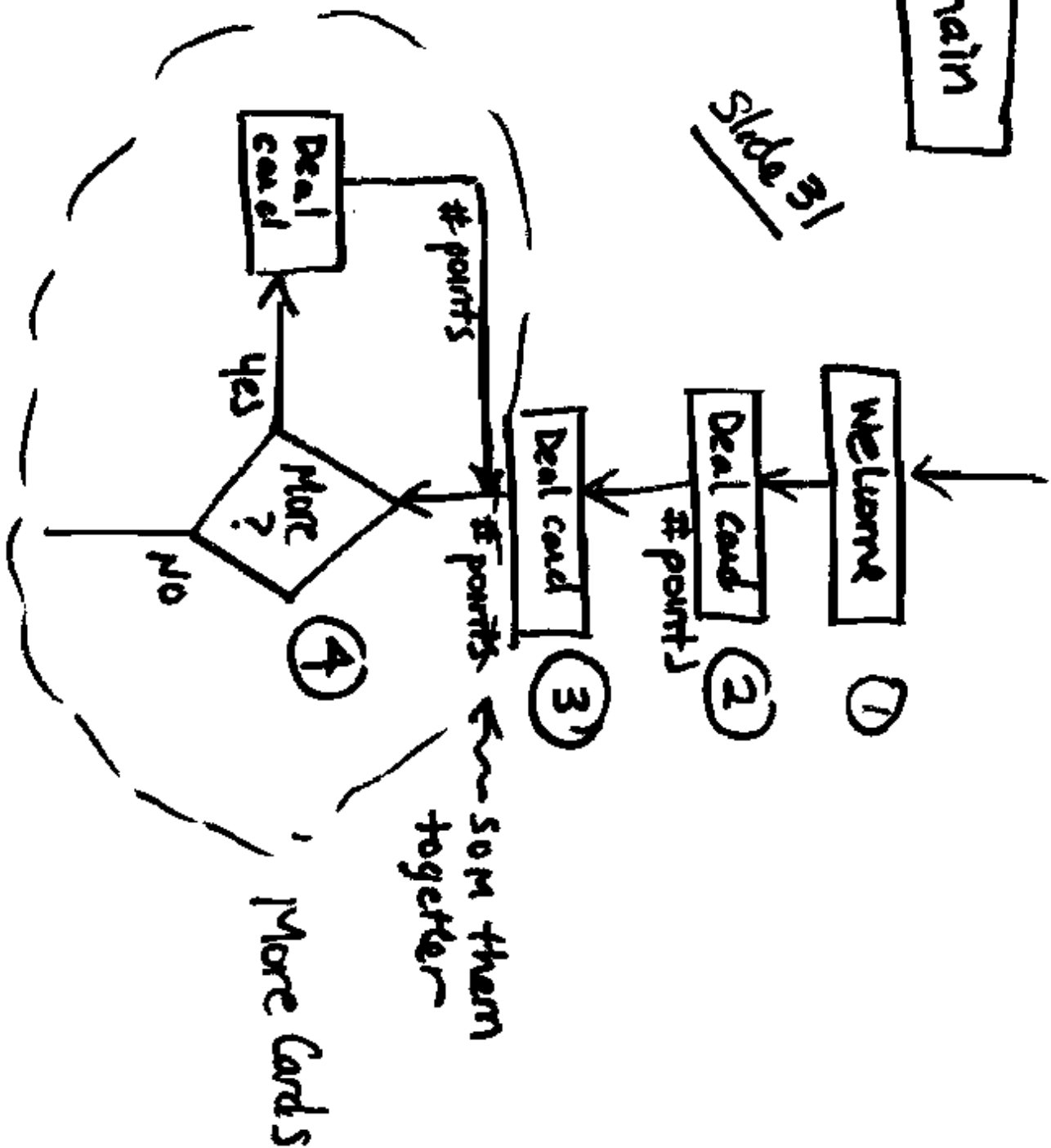
```
    card = rand() % 13 +1;    //Make sure the card is within range 0-13
```

```
    if (1 == card)    //ACE!!
```

- We have seen how to deal one card, but...
 - We need to start off with the player getting two cards and
 - Then getting “hit” as many times as they want.
 - We don't need to change the deal_card function but rather just call it a few more times.
 - Our “algorithm” will be:
 - For the player
 - Deal a card storing the points
 - Deal another card and add those points to our running total
 - Ask the user if they want to be “hit”
 - If so, deal another card and add those points to the running total
 - Continue to do so until the user is happy or until the points are over 21

main

Slide 3!



```
int main()
{
    int user_points; //to hold the number of points the player has
    char hit; //do they want to be hit?
    srand(time(0));
```

① welcome(); //calling the function to display the rules

//deal the players hand

② user_points = deal_card(); //Deal a card (FIRST)
user_points += deal_card(); //Deal the second card
do

{ //Does the user want a "hit"?

④ cout << "Do you want another card? y/n ";

cin >> hit; cin.get();

if ('y' == hit || 'Y' == hit)

十三

user_points += deal_card();

} while ((hit == 'y' || hit == 'Y') && user_points < 21);

↑ ↑

cout << "You have " << user_points << " points" << endl;

if (user_points > 21)

cout << "You lost! " << endl;

cin.get();

return 0;

}

③

user-points

↓ = deal-card()

user-points = user-points + deal-card()



main

welcome

deal card

points

deal card

points

More cards

points

> 21 ?

yes

no

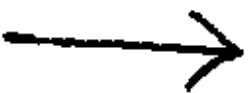
LOST!

Sum them together

Sum them together

// prototype

int more_cards();



#points

φ ae ?



```
void welcome ();    //This is where the rules will be described
int deal_card();    //get a card and find out the points...
int more_cards();   //returns the points accumulated
int main()
{
    int user_points; //to hold the number of points the player has
    srand(time(0));
    welcome();      //calling the function to display the rules

    //deal the players hand
    user_points = deal_card(); //Deal a card
    user_points += deal_card(); //Deal the second card
```

```
user_points += more_cards(); //keep getting cards
```



```
cout <<"You have " <<user_points <<" points" <<endl;
```

```
if (user_points > 21)
```

```
    cout <<"You lost!" <<endl;
```

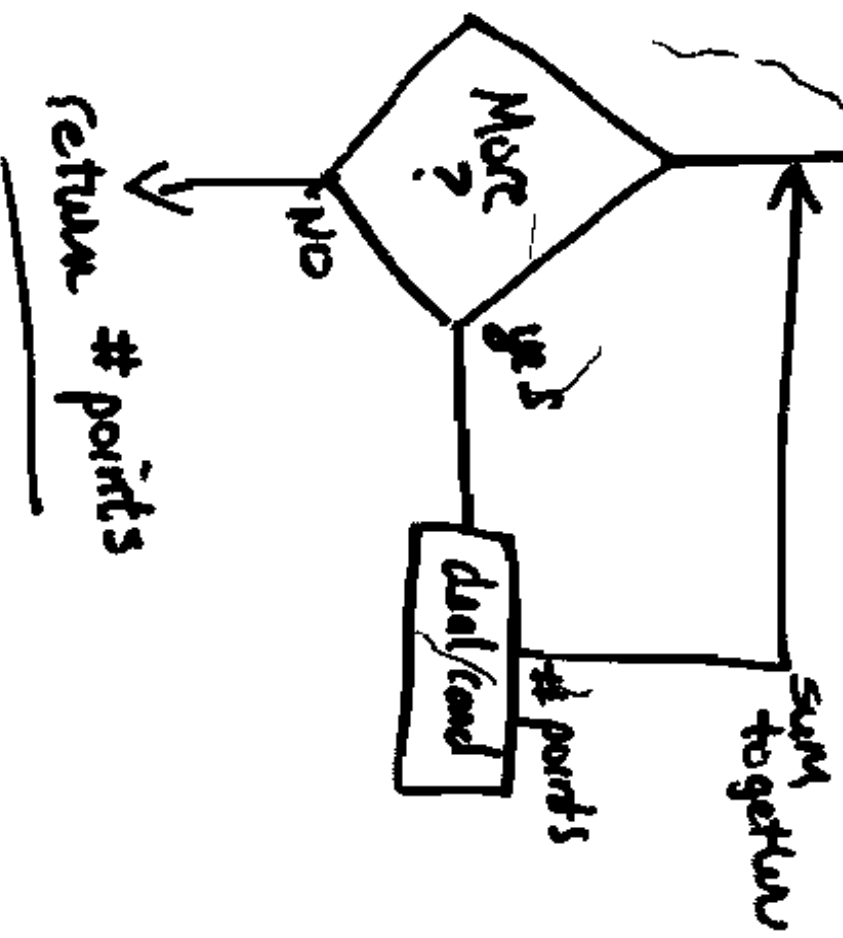
```
    cin.get();
```

```
    return 0;
```

```
}
```


More Cards

Assumption: #points is zero



```
//Continue to deal cards until the user is happy
// or the value is too great
```

```
int more_cards()
{
    char hit;
    int points = 0;
    do
    {
        cout <<"Do you want another card? y/n ";
        cin >>hit;  cin.get();
        if ('y' == hit || 'Y' == hit)
            points += deal_card();
    } while ((hit == 'y' || hit == 'Y'));
}
return points;
}
```

Main

Welcome

deal card

points

deal card

points

More cards

points

> 21

yes

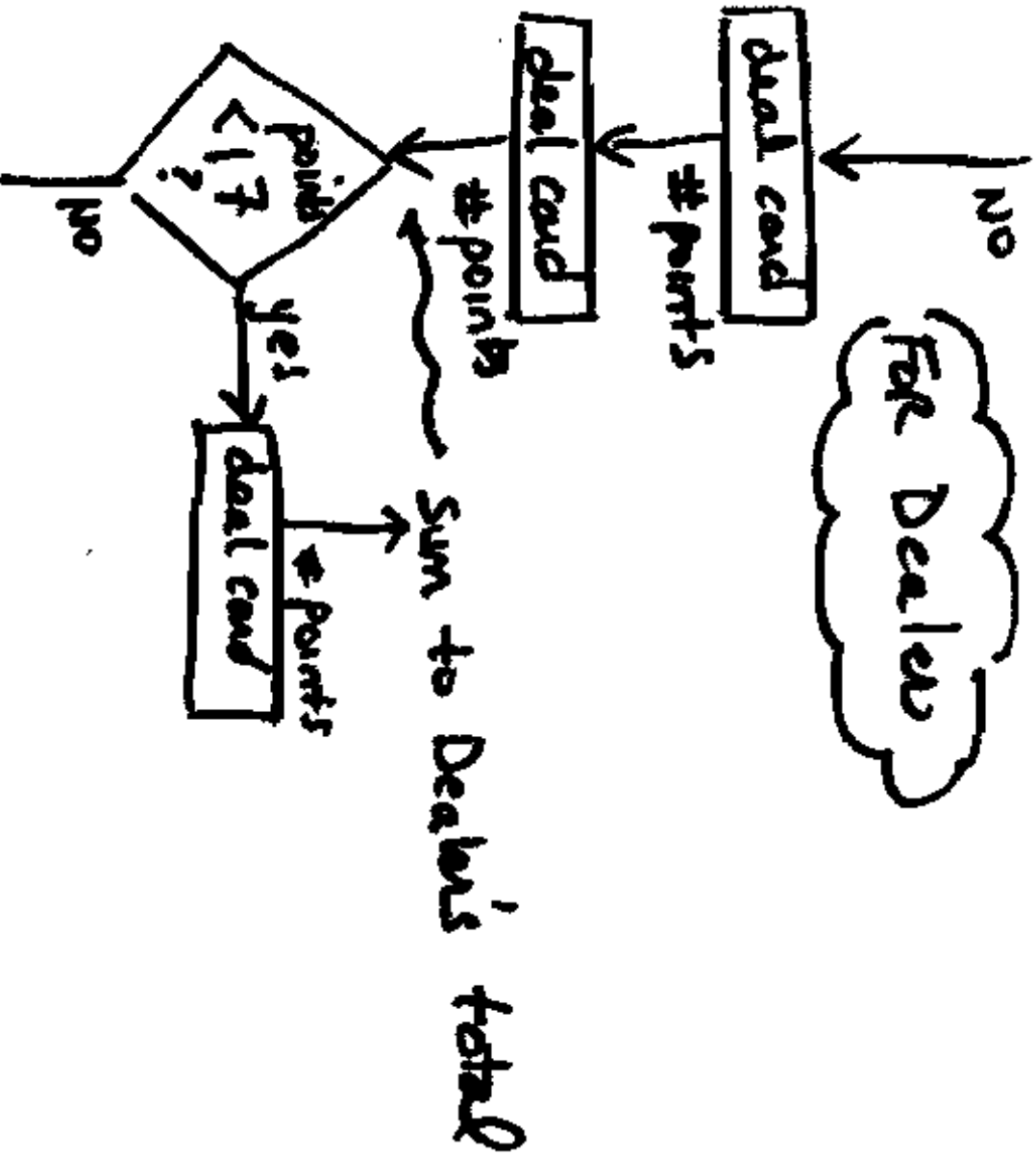
LOST!

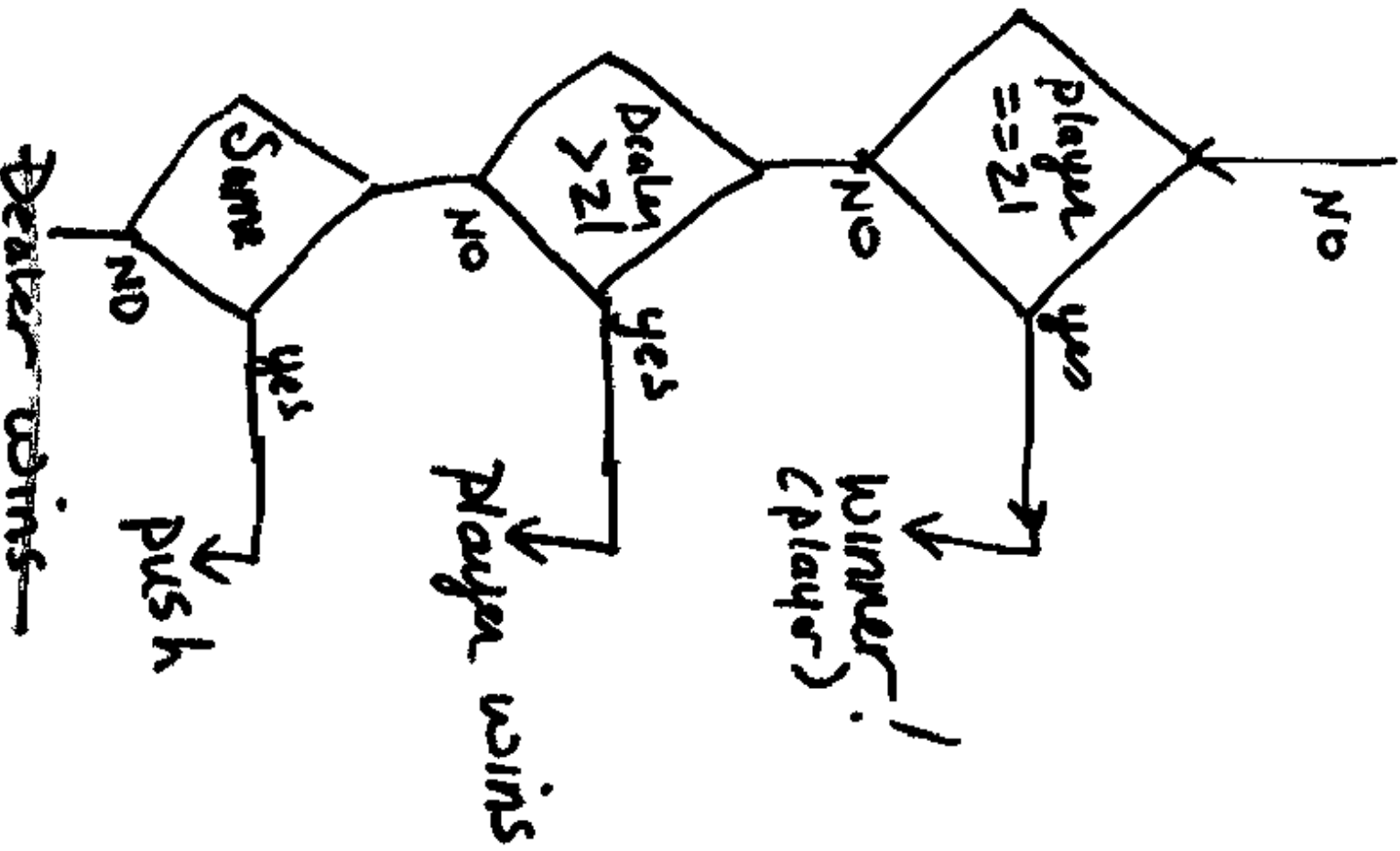
For Player

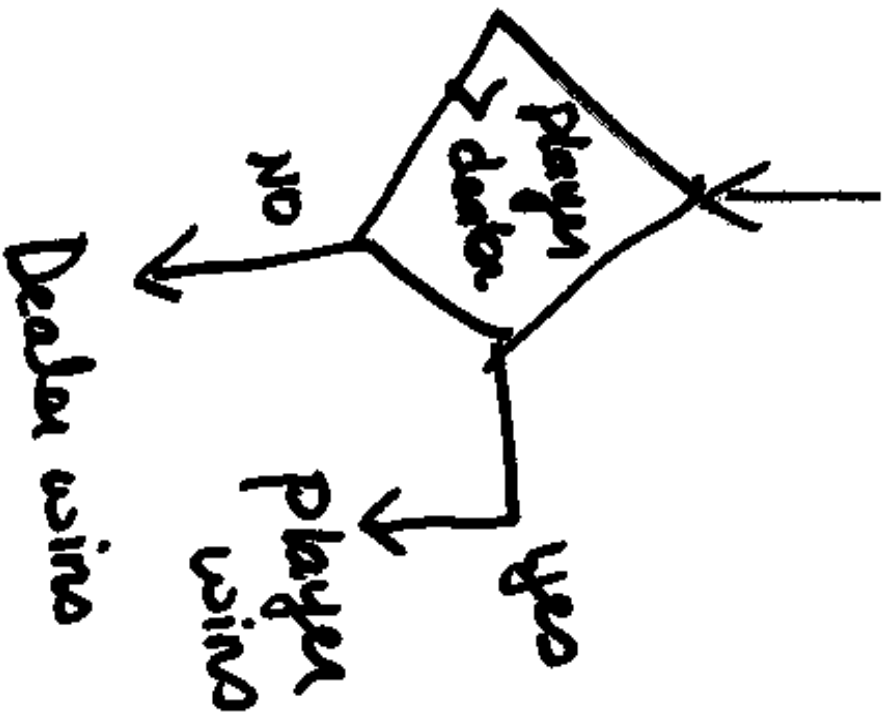
sum player's total

sum player's total

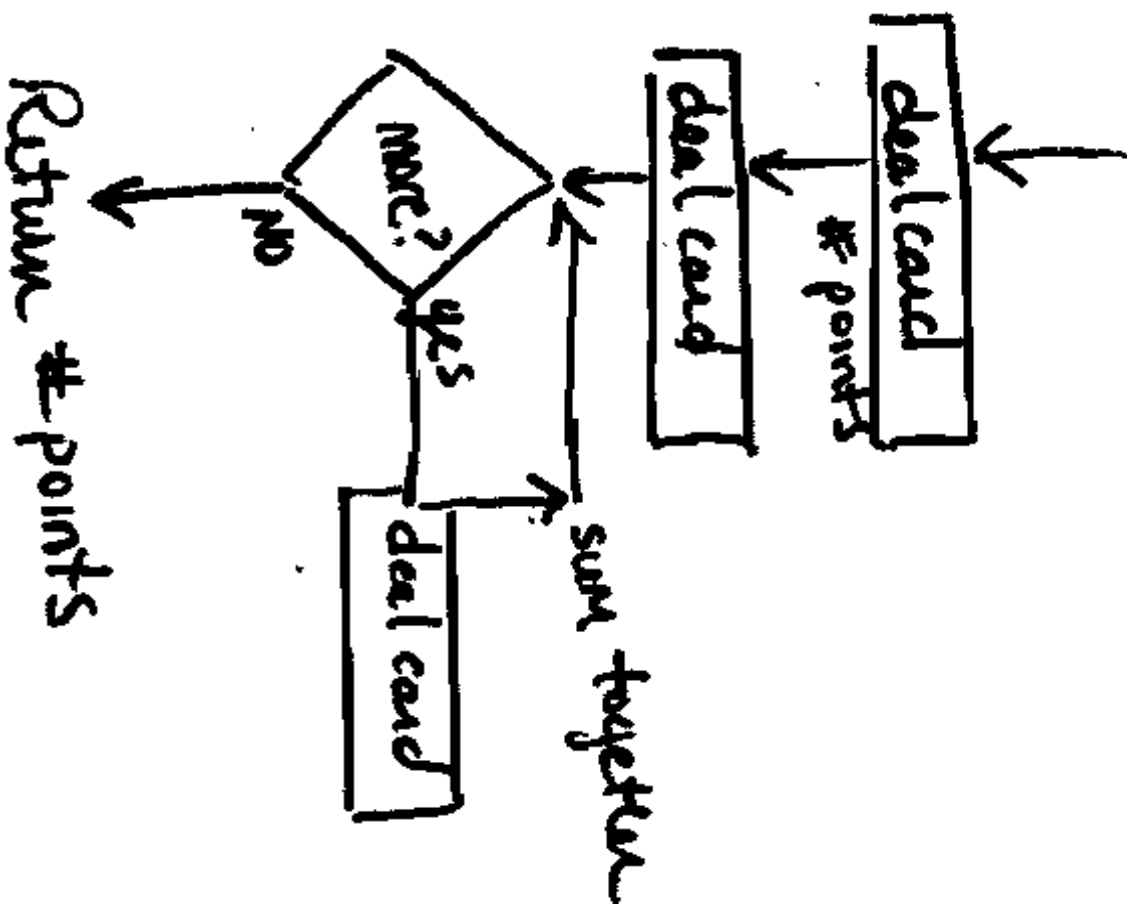
For Dealer







Player



```
//deal the players hand  
int player()
```

```
{
```

```
int points;
```

```
//accumulate points for this player
```

```
cout <<"IT IS NOW THE PLAYERS TURN:" <<endl;
```

```
points = deal_card(); //Deal a card
```

```
points += deal_card(); //Deal the second card
```

```
points += more_cards(); //keep getting cards until 21
```

```
cout <<"You have " <<points <<" points" <<endl;
```

```
return points;
```

```
}
```


Dealer

Deal card

Deal card

< 17

yes

deal card

sum together

No

Return # points

//Slide 52

//deal the dealer's hand

int dealer()

{

int points;

cout <<endl <<endl <<"IT IS NOW THE DEALERS TURN: " <<endl;
points = deal_card();
points += deal_card();

//dealer's must accept more cards if their total is less than 17

while (points < 17)

points += deal_card();

return points;

main

Welcome

Player

points for player

Dealer

points for dealer

Winner

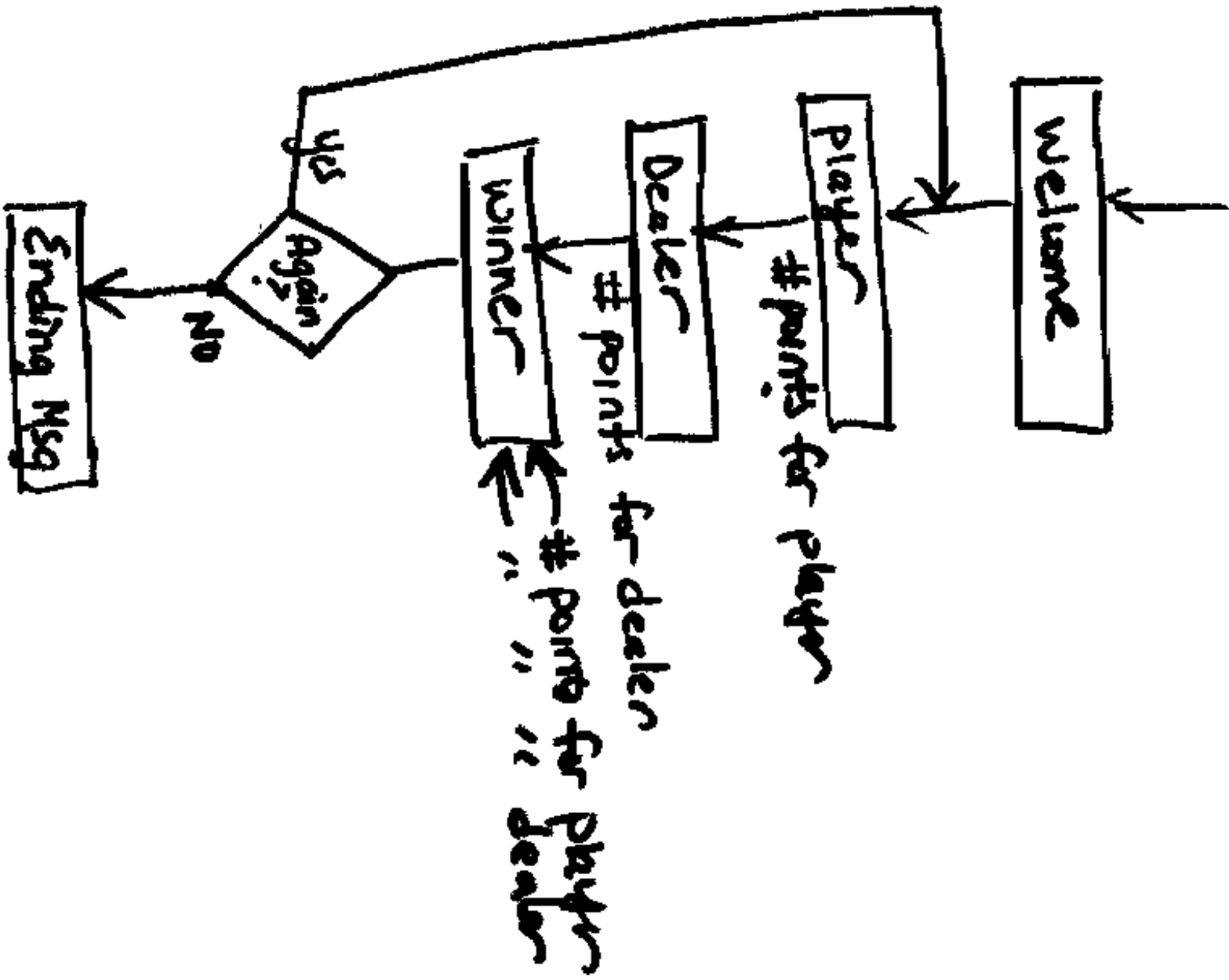
points for player
points for dealer

Again?

yes

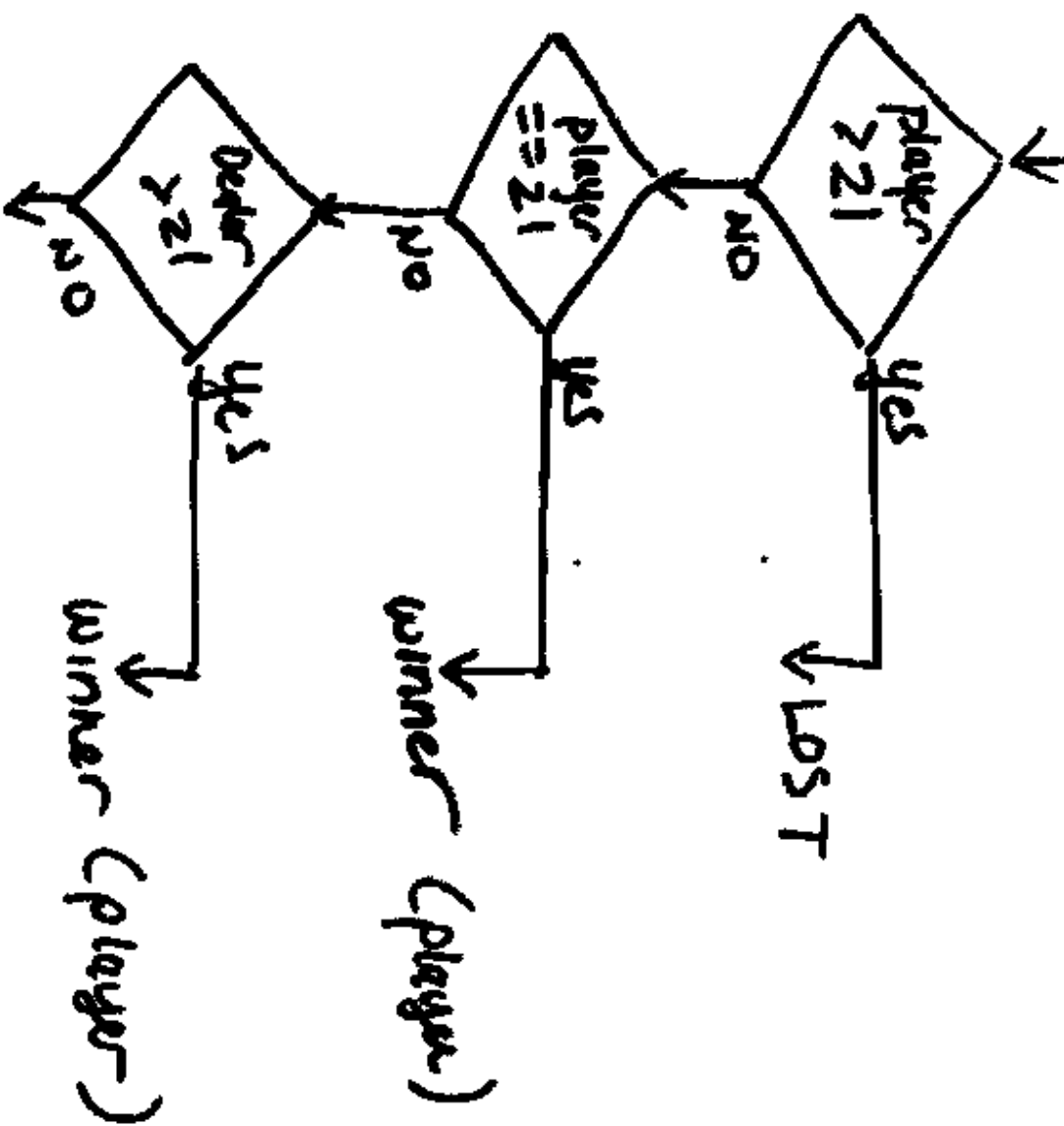
NO

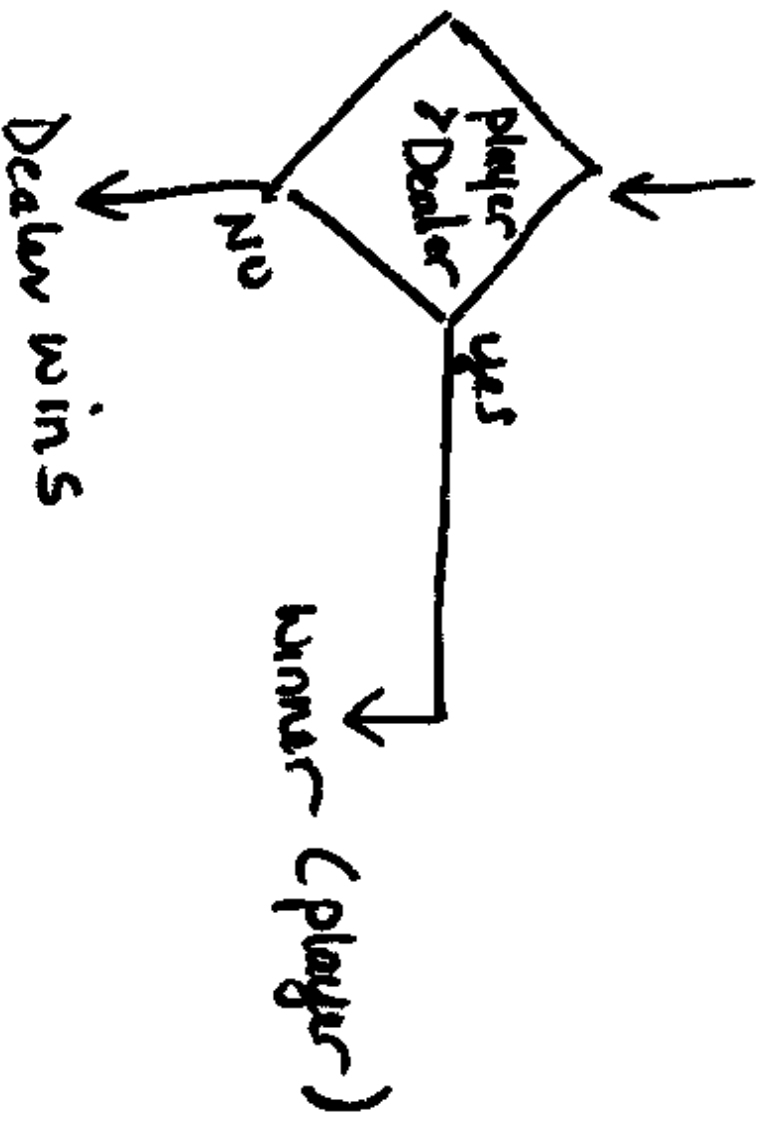
Ending Msg



winner

points for player
points for dealer





```
//slide 57
/Display a message as to which player wins
//Return true if the player doesn't win
void winner(int player, int dealer)
{
    //First let's see if the player is the clear winner
    if (player > 21)
        cout <<"You lost!" <<endl;
    else if (player == 21)
        cout <<"You won!" <<endl;

    //now let's see how the dealer did!
    else if (player > dealer || dealer > 21)
        cout <<"You beat the Dealer!! Great job" <<endl;
    else if (player == dealer)
```

```
cout << "Push - try again next time" << endl;
else
    cout << "Better luck next time - Dealer Rules!" << endl;
```

```
}
```

main

Welcome

Player

player points

Dealer

Dealer points

winner



yes



no

//Slide 64

bool play_again(); //Do you want to play again?

int deal_card(int dealer=-1); //get a card and find out the points...

int main()

{

int user_points; ~~It~~to hold the number of points the player has

int dealer_points; //to hold the dealer's points

srand(time(0));

welcome(); //calling the function to display the rules

do //let's allow the user to play as much as they want

{

user_points = player();

dealer_points = dealer();

```
winner{user_points, dealer_points}; //call  
} while (play_again()); //slide 65  
ending_message(); cin.get();  
return 0;  
}
```