Michał Kucharski

Środa 12:15

Projekt nr 2

**1.Treśc zadania.**

Napisać program umożliwiający grę w Black Jacka z komputerem.

Program ma opierać się o tablice.

Należy zapewnić czytelną komunikację z użytkownikiem(interfejs, prezentacja wyników).

Konieczne jest zabezpieczenie programów przed podaniem niepoprawnych danych.

**2.Propozycja rozwiązania**

* Wylosowanie dwóch kart dla użytkownika i jednej dla komputera.
* Kolejka użytkownika (dobieranie kart lub spasowanie)
* Kolejka komputera (dobieranie kart aż do 17 i jeśli przekroczy to pas)
* Gdy suma punktów gracza przekroczy 21, ale któryś z nich właśnie dobierze asa to od sumy odejmowane jest 10 punktów i gra toczy się dalej.
* Porównanie sumę punktów użytkownika i gracza, a następnie wypisanie na ekran kto wygrał

**3.Zmienne i struktury**

* Zmienne typu int
* Tablica kart gracza (karty\_gracza)
* Tablica kart komputera (karty\_krupiera)
* Suma punktów gracza (suma\_gracza)
* Suma punktów krupiera (suma\_krupiera)
* Iterator (i)
* Wybór 1 lub 2 (choice)
* Liczba punktów danej karty (moc)
* Stale zmienne deklarowane przy użyciu #define
* Oczko 21 (OCZKO)
* Wymiary tablicy kart graczy 10 (MAX\_CARDS)
* Maksymalna wartość sumy krupiera po której ma spasować 17 (KRUPIER\_MAX)
* Liczba kart 52 (SIZE)
* Wskaźnik na int
* Wskaźnik na sumę gracza (gracz\_ptr)
* Wskaźnik na sumę krupiera(krupier\_ptr)

**4. Schemat blokowy algorytmu**

START

Czy gracz chce dobrać?

TAK

Dobranie karty gracza

Dodanie mocy karty do sumy gracza

NIE

Dobranie karty krupiera

Dodanie mocy karty do sumy krupiera

TAK

Suma\_krupiera < 17

NIE

Porównanie sum i wypisanie zwyciezcy

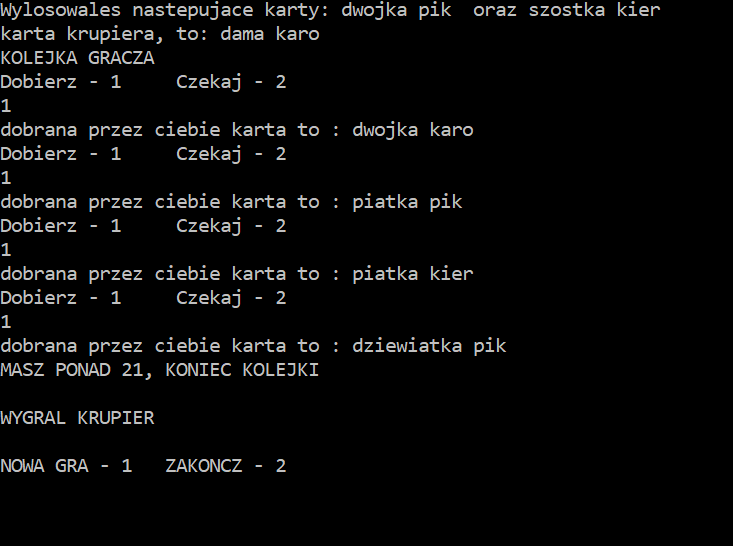
KONIEC

**5.Funkcje**

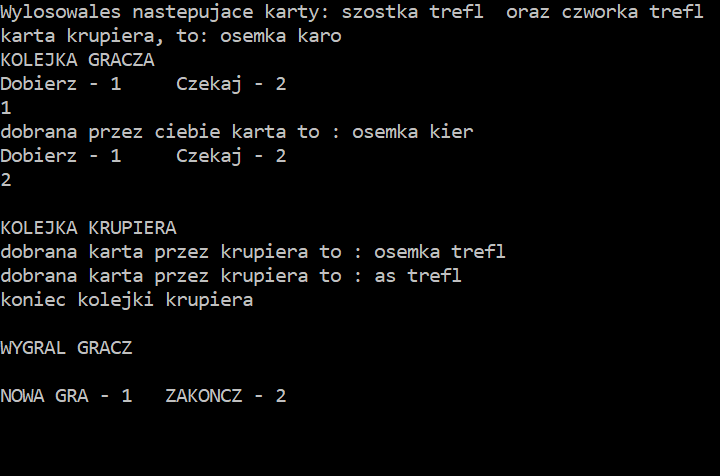
* losowanie
* Zwraca wylosowaną liczbę z zakresu 0-51
* Void dobranie
  + Używa jako parametrów sumy gracza, wskaźnika na karty gracza, liczby całkowitej która jest indeksem w tablicy kart gracza oraz wskaźnika na karty przeciwnika
  + W pętli wywołuję funkcję losowanie
  + Sprawdza czy dana karta nie została już wylosowana(wtedy wykonuje kolejną iterację), a jeśli nie to wychodzi z pętli
  + Przy użyciu funkcji jaka\_karta dodaje moc danej karty do sumy gracza
* Int jaka\_karta
* Jako parametru używa liczby całkowitej
* Przy pomocy instrukcji switch case określa, która karta jest przypisana do danej liczby całkowitej
* Wyświetla nazwe danej kart i zwraca przypisane jej punkty
* Void rozdanie
* Wywołuje funkcję dobranie dwukrotnie dla sumy gracza i kart gracza
* Wywołuje funkcje dobranie raz dla sumy krupiera i kart krupiera
* Void kolejka\_gracza
  + Używa jako parametrów wskaźnika na karty gracza, wskaźnika na sumę gracza i wskaźnika na karty krupiera
  + W pętli daje wybór dobierz lub czekaj i w zależności od wyboru wywołuje funkcję dobranie lub wychodzi z pętli
  + Wychodzi z pętli również wtedy gdy suma\_gracza przekroczy 21 pkt., chyba, że dobrana karta to as, to wtedy redukuję sumę o 10pkt.
* Void kolejka\_krupiera
* Używa jako parametrów wskaźnika na karty krupiera, wskaźnika na sumę krupiera i wskaźnika na karty gracza
* W pętli wywołuje funkcje dobranie
  + Wychodzi z pętli gdy suma krupiera przekroczy 17, chyba że dobrana karta to as, to wtedy redukuję sumę o 10pkt.
* Void wygrany
  + Używa jako parametrów suma gracza i sumy krupiera
  + Na podsatwie sum gracza i krupiera wypisuje na ekran kto wygrał
* Int sprawdzenie
* W pętli pobiera zmienną i sprawdza czy jest to liczba całkowita 1 lub 2
* Jeśli jest to 1 lub 2 to funkcja zwraca pobraną zmienną
* Jeśli jest to coś innego niż 1 lub 2 to wykonuję się następna iteracja

**6. Testowanie**

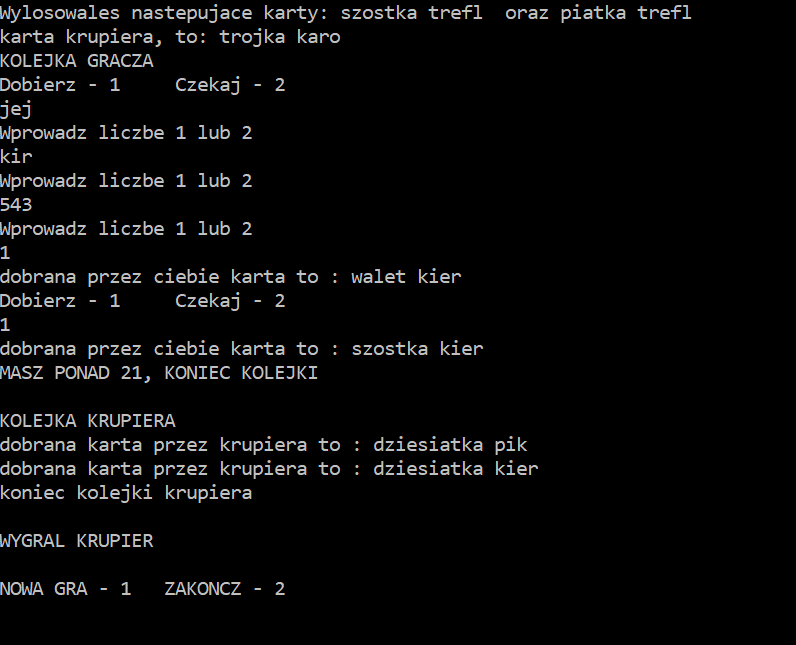
* Wygrana krupiera



* Wygrana gracza



* Wpisanie liter lub innych liczb niż 1 lub 2



**7. Odstępstwa**

1. Maksymalna liczba kart które może dobrać gracz to 8 a krupier 9
2. Krupier dobiera karty do 17
3. Po przekroczeniu 21 pkt kolejka gracza się kończy i wygrywa krupier
4. Może wygrać gracz, może wygrać krupier lub może być remis
5. Gra odbywa się jedną talią kart