**ZADANIE 1 – GRA W OCZKO**

***Zadanie zaproponował: mgr Krzysztof Jarczewski, III LO im. S. Batorego w Chorzowie***

W sposób losowy układamy liczby naturalne od 1 do 11, które wprowadzamy z „klawiatury” lub są zapisywane do tablicy G.

Gra polega na tym, że gracz A dobiera kolejne liczby z tablicy G, tak aby otrzymać sumę najbliższą liczbie 21, ale jej nie przekroczyć (**przegrana**). Gdy gracz A rezygnuje z doboru kolejnej liczby, gracz B dobiera liczby z tablicy G.

Gracz A wygrywa, gdy jego suma nie przekroczy liczby 21 i będzie większa lub równa sumie gracza B. Gracz B wygrywa, gdy jego suma nie przekroczy liczby 21 i będzie większa od gracza A.

Twoim zadaniem jest napisanie programu, który:

* umożliwia wybór wprowadzania liczb:

a) z klawiatury,

b) w sposób losowy zapisuje liczby do tablicy G,

* symuluje powyższą grę w „**oczko**”,

1. podaje ilość możliwości przekroczenia sumy 21 przy doborze kolejnej liczby przez gracza A,
2. podaje ilości możliwości otrzymania sumy 21 – „**oczko”** przez gracza A przy uzyskaniu przez niego sumy większej niż 10,
3. przy uzyskaniu przez gracza A sumy większej niż 14, oblicza ilości możliwości **wygranej** gracza B, czyli otrzymania przez gracza B sumy większej niż obecna suma gracza A oraz nie większej niż 21.

Przykłady.

1. Wprowadzone liczby z klawiatury dla gracza A: 2, 3, 4, 5, 6.

Obliczenia:

ilość możliwości otrzymania sumy 21: 1, {1},

ilość możliwości otrzymania sumy większej niż 21: 5, {7, 8, 9, 10, 11}

ilość możliwości wygrania gry przez gracza B: 6. {(1,9,11), (1,11,9), (9,1,11), (9,11,1), (10,11), (11,10)}

1. Wprowadzone liczby z klawiatury dla gracza A: 1, 5, 8.

Obliczenia:

ilość możliwości otrzymania sumy 21: 3, {(3,4), (4,3), (7)},

ilość możliwości otrzymania sumy większej niż 21: 3, {9, 10, 11}.

1. Wprowadzone liczby z klawiatury dla gracza A: 5, 3, 1.

Obliczenia:

ilość możliwości otrzymania sumy większej niż 21: 0.

**ZADANIE 2 – ZADANIE NA CZASIE**

***Zadanie zaproponował: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska***

Wskazówki zegara o dowolnej godzinie tworzą parę kątów (najczęściej różnych). Np. o godzinie 12.00 kąty te mają miary i , o godzinie 18.00: i , a o godzinie 15.00: i .

Napisz program, który dla zadanych argumentów i , odpowiadających godzinie i minucie (dokładność sekundową pomijamy, przyjmując, że wskazówka sekundowa znajduje się na „12”), gdzie oraz , zwracał będzie nie większy z pary tych kątów wyrażony w radianach (symbol może być wypisywany słownie).

W powyższych przykładach mielibyśmy odpowiednio:

– dla godziny 12.00 argumenty to , , program zwraca 0;

– dla godziny 18.00 argumenty to , , program zwraca ;

– dla godziny 15.00 argumenty to , , program zwraca .

**ZADANIE 3 – OSTATNIA CYFRA**

***Zadanie zaproponował: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska***

Rozpatrzmy sumę:

gdzie są kolejnymi cyframi systemu dziesiątkowego, a są liczbami naturalnymi.

Napisz program, który dla zadanej listy dziesięciu liczb naturalnych , zwracał będzie ostatnią cyfrę powyższej sumy .

Przykładowo, zadając kolejno (lub jako tablicę – sposób wprowadzania wartości potęg pozostawiamy rozwiązującemu): 3, 4, 78, 9, 1, 12, 3, 2, 10, 5, ostatnią cyfrą sumy

jest cyfra 5, którą to cyfrę zwróci program.

Pamiętaj o tym, że bardzo łatwo w tym zadaniu przekroczyć zakres odpowiedniej zmiennej. Miej to na uwadze pisząc ten program (nie używaj specjalnych bibliotek dla dużych liczb).

**ZADANIE 4 – SUMA ODWROTNOŚCI SUM**

***Zadanie zaproponował: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska***

Napisz program, który dla zadanej liczby naturalnej , zwracał będzie wartość sumy:

Za wynik numeryczny przyznawany jest jeden punkt, a za wynik w postaci numerycznej i symbolicznej (sposób zwracania wyniku symbolicznego pozostawiamy rozwiązującemu) komplet punktów.

**ZADANIE 5 – TATARAKI I BALONIK**

***Zadanie zaproponował: dr inż. Mariusz Pleszczyński, Wydział Matematyki Stosowanej, Politechnika Śląska***

W pliku *słownik.txt* znajduje się słownik, w którym słowa (każde w nowej linii) posortowane są alfabetycznie.

Napisz program, który dla zadanej początkowej litery słowa, poszukiwał będzie w tym słowniku wyrazów, które da się podzielić na pewnym miejscu w ten sposób, że zarówno do miejsca podziału jak i od miejsca podziału (tę część czytamy wspak), tak powstałe słowa, również znajdują się w tym słowniku.

Przykładowo, jeśli podalibyśmy jako argument literę *t*, to program mógłby znaleźć słowo *tataraki*, bo dzieląc je po czwartej literze, otrzymamy słowa *tata* i *ikar*, a dla litery *b*, program mógłby zwrócić słowo *balonik* (podział po trzeciej literze na słowa *bal* i *kino*).

Zakładamy dodatkowo, że zarówno poszukiwane słowo, jak i jego składowe, są co najmniej dwuliterowe.