Laboratorium 7

Funkcje I, II

Zadanie 1.

Napisz funkcję do sprawdzania czy liczba jest doskonała. Funkcja dla podanej przez użytkownika liczby ma sprawdzić czy jest ona doskonała i zwrócić 1 jeśli tak i 0 jeśli nie. Użyj jej do sprawdzenia i wypisania dla 100 pierwszych liczb czy każda z nich jest doskonała czy nie.

```
int czy_doskonala (int liczba)
```

Zadanie 2.

Napisz funkcję **next_3n_1**(), która przy każdym wywołaniu zwróci kolejną liczbę z ciągu Collatz'a *3n+1*. Funkcja przyjmuje jeden parametr, jeżeli przekazana do funkcji wartość jest równa 0, to funkcja powinna zwrócić kolejny element ciągu, jeżeli do funkcji zostanie przekazania liczba dodatnia, to funkcja powinna zacząć generować elementy ciągu dla tej liczby.

```
int next_3n_1 (int start)
```

Zadanie 3.

Napisz funkcję do zerowania wszystkich elementów wektora podzielnych przez „***number***" i zwracająca ilość wyzerowanych elementów.

```
int clear_vector (int tab[], int size, int number)
```

Przykładowa interakcja:

```
int tab[10] = {2, 3, 2, 3, 2, 3, 2, 3, 2, 3};
printf ("liczba wyzerowanych to %d\n", clear_vector (tab, 10, 2));
```

Odpowiedź po kompilacji:

```
liczba wyzerowanych to 5
```

Zadanie 4.

Napisz funkcję do sprawdzania czy podany numer PESEL jest prawidłowy.

Funkcja ma przyjąć char **pesel**[] jako parametr wejściowy i zwrócić **1** przypadku poprawności peselu **0** niepoprawności i **-1** w przypadku błędnych danych.

```
int check_PESEL (const char tab[])
```

ALGORYTM WALIDACJI NUMERU PESEL

Numer PESEL jest to 11-cyfrowy, stały symbol numeryczny, jednoznacznie identyfikujący określoną osobę fizyczną.
Zbudowany jest z następujących elementów:

- *cyfry [1-6] - data urodzenia w postaci: rrmmdd*
- *cyfry [7-9] - liczba porządkowej*
- *cyfra [10] - płeć (cyfry parzyste oznaczają płeć żeńską, nieparzyste) - męską)*
- *cyfra [11] - cyfra kontrolna*

Dla odróżnienia poszczególnych stuleci przyjęto następującą metodę kodowania:

- *dla lat 1900 do 1999 - miesiąc zapisywany jest w sposób naturalny*
- *dla lat 1800-1899-80: miesiąc = miesiąc + 80*
- *dla lat 2000-2099-20: miesiąc = miesiąc + 20*
- *dla lat 2100-2199-40: miesiąc = miesiąc + 40*
- *dla lat 2200-2299-60: miesiąc = miesiąc + 60*

Algorytm weryfikacji numeru PESEL jest następujący:
Każda z pozycji numeru ewidencyjnego posiada stały współczynnik zwany wagą pozycji.
Każdą cyfrę numeru ewidencyjnego mnożymy przez odpowiednią wagę:

1-3-7-9-1-3-7-9-1-3

Sumujemy wyniki mnożenia.
Otrzymany wynik dzielimy modulo 10 i odejmuje od 10.
Jeżeli wynikiem jest 10, to przyjmujemy 0.
Otrzymany wynik jest cyfrą kontrolna numeru PESEL i powinien być równy jego ostatniej cyfrze.

Zadanie 5.

Napisz program, który będzie miał funkcje do wypisywania znaków, które istnieją w tablicy **tab1**[] a nie ma ich w tablicy **tab2**[]. Poproś użytkownika o pierwszy i drugi string i użyj tej funkcji do pokazania różnic w znakach. Funkcja ma zwracać **1** w przypadku jeśli istniał choć 1 znak różniący się, a **0** w pozostałych przypadkach.

```
int show_differences (const char tab1[], const char tab2[])
```

Przykładowa interakcja:

```
char tab1[] = "abxyas";
char tab2[] = "abx";
```

Odpowiedź po kompilacji:

```
Brak znaku y
Brak znaku s
```

Zadanie 6.

Dla talii kart od 2 do Asa (przyjmij liczby dla kart nienumerycznych) napisz funkcje, które będą losować karty dla gracza, wypisywać jego karty w przyjaznej formie oraz funkcję do grania w „mocniejsze słabsze" – mocniejsza karta zdobywa punkty w przypadku remisu punktu nie dostaje nikt.

Funkcja **graj**() ma wyświetlić podsumowanie punktów, które zdobyli poszczególni gracze.

Wskazówka: Nie musimy brać pod uwagę, że rozlosowane zostanie więcej niż 4 karty danego typu np. Króli.

```
void losuj_karty (int tab[])
void wypisz_karty (const int tab[], int id_gracza)
void graj (const int player1[], const int player2[])
```

Przykładowa interakcja:

```
Karty gracza numer 1
Karta 1 to 2
Karta 2 to 9
Karta 3 to 10
Karta 4 to 3
Karta 5 to 7
Karta 6 to 3
Karta 7 to 3
Karta 8 to KROL
```

```
Karty gracza numer 2
Karta 1 to 2
Karta 2 to KROL
Karta 3 to 6
Karta 4 to 3
Karta 5 to DAMA
Karta 6 to 5
Karta 7 to DAMA
Karta 8 to WALET

Wyniki gracz1 to 2, gracz2 to 4
```

Dla takiej kolejności:

```
int player1[CARDS]={0};
int player2[CARDS]={0};
losuj_karty(player1);
losuj_karty(player2);
wypisz_karty(player1,1);
wypisz_karty(player2,2);
graj(player1,player2);
```