PROGRAMOWANIE STRUKTURALNE KOLOKWIUM 1
MAJ 2020 - ZDALNE

Zadanie 1.
Popraw kod, tak aby spelniał on zasady kompilacji, a program zwracał wyniki zgodne z prawdą. Wolno zmienić tylko tą część kodu, która znajduje się przed mainem.

KOD DO POPRAWIENIA - ZAŁĄCZNIK DO ZADANIA 1

```
int modul(int const x)
{
    if(x<0)
{
    x=-x;
}
    return x;
}

void podwoj(int x)
{
    return 2*x;
}

int kwadratowa(int x)
{
    return x*x;
}

float odwrotnosc(unsigned int x)
{
    return 1/x;
}

float potegaodwrotnosci(unsigned int x)
{
    return pow(2,odwrotnosc(x));
}

int main()
{
    int y=-3;
    printf("Wartosc bezwzgledna liczby %d wynosi %d\n",y,modul(y));
    int z;
    printf("Podaj liczbe z\n");
    scanf("%d",&z);
printf("Dwukrotnosc liczby %d wynosi %d\n",z,podwoj(z));
int w=5;
printf("Wartosc funkcji kwadratowej na argumencie %d wynosi %f\n",w,kwadratowa(w));
int u=2;
printf("Dwa do potegi 1/%d wynosi %f",u,potegaodwrotnosci(u));
    return 0;
}
```

Zadanie 2.

a) Napisz program, który obliczy sumę wszystkich liczb naturalnych trzycyfrowych podzielnych przez 5, ale nie podzielnych przez 3.

b) Napisz funkcję int cyfra_wiodaca(int x), która ma zwracać pierwszą od lewej cyfrę w zapisie dziesiętnym liczby x, podanej w agrumencie. Np. cyfra_wiodaca(8645)=8, cyfra_wiodaca(59)=5, cyfra_wiodaca(4)=4.
Należy wykorzystać funkcję pomocniczą, liczącą z ilu cyfr składa się liczba, np. ile_cyfr(8645)=4, ile_cyfr(59)=2, ile_cyfr(4)=1. W jej implementacji kluczowym jest porównywanie agrumentu z kolejnymi potęgami liczby 10. Np. $8645 \geq 10^4$, ale już $8645 < 10^5$, więc dlatego składa się z 4 cyfr. Gdy już ustalimy rząd wielkości za pomocą liczby cyfr, porównujemy argument z jego kolejnymi wielokrotnościami np. $8645 \geq 1000, 8645 \geq 2000, \ldots, 8645 \geq 8000$, ale $8645 < 9000$, więc cyfrą wiodącą jest 8.

c) Wśród wielomianów stopnia 3 (czyli funkcji postaci $W(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$) wiadomo, że istnieje taki o współczynnikach całkowitych, nie większych co do wartości bezwzględnej od 10, który dla argumentów $0, 1, 2, 3$ przyjmuje kolejno wartości $6, 8, 18, 54$. Znajdź ten wielomian.

Zadanie 3.

a) Napisz program, który wyznaczy n-ty wyraz ciągu, którego pierwszy wyraz to 1, a każdy kolejny jest o 1 mniejszy od trzykrotności poprzedniego. Wypisz pierwsze 10 wyrazów tego ciagu.

b) Napisz program, który zbada ciąg zdefiniowany w następujący sposób: pierwszy wyraz jest liczbą naturalną, a każdy kolejny jest połową poprzedniego, w przypadku gdy poprzedni był parzysty albo liczbą o 1 wiekszą od poprzedniego, gdy poprzedni był nieparzysty. Program ma poprosić użytkownika o określenie pierwszego wyrazu, który ten poda za pomocą operacji standardowego wejścia, a następnie wypisać wszystkie wyrazy tego ciagu, aż do momentu gdy któryś z nich bedzie równy 1.
Np. gdy użytkownik poda 10 program powinien wypisać $10, 5, 6, 3, 4, 2, 1$.

c)Funkcja int dwie_zmienne(int n, int m) zdefiniowana jest w następujący sposób:
dwie_zmienne(0,0)=1 oraz dwie_zmienne(1,0)=1;
dwie_zmienne(n+2,0)=dwie_zmienne(n,0)+dwie_zmienne(n+1,0) dla wszystkich naturalnych n;
dwie_zmienne(n,m+1)=dwie_zmienne(n,m)+2· dwie_zmienne(n+1,m) dla dowolnych liczb naturalnych n,m;
Wypisać na standardowym wyjściu wartość dwie_zmienne(3,4).

Zadanie 4.

a)Zdefiniować stałą typu float. Wyświetlić jej adres.

b)Zarezerowować pamięć na zmienną typu int. Przypisać wartość zmiennej pod tym adresem na 17.

c) Napisz funkcję, której argumentami są trzy wskaźniki na zmienną typu int, która zwraca wskaźnik na środkową co do wielkości w porządku rosnącym ze wskazywanych wartości w przypadku gdy są one rożne od siebie;
w przypadku gdy dwie spośród wskazywanych zmiennych mają równe wartości, ma zwracać wskaźnik na zmienną, której wartość nie jest im równa, a w przypadku gdy wszystkie trzy wskazywane wartości są równe, wskaźnik na dowolną;
Np. gdy wartości wskazywane będą równe $4, 7, 5$ zwrócony ma zostać wskaźnik na 5;
Np. gdy wartości wskazywane będą równe $3, 3, 6$ zwrócony ma zostać wskaźnik na 6;

d) Napisz funkcję o wartości typu bool, której argumentami są zmienna typu unsigned int n i dwa wskaźniki na funkcje typu int o jednym argumencie typu unsigned int. Funkcja ma sprawdzać, czy funkcja wskazywana przez drugi wskaźnik ma wartość równą kwadratowi funkcji wskazywanej przez pierwszy wskaźnik na argumencie równym n. Przetestować na wybranych przez siebie funkcjach i liczbie n, tak aby przynajmniej raz zwrócona była każda z wartości true i false.

Zadanie 5.

a) Napisz program, który wypisze wszystkie elementy tablicy o indeksach parzystach w odwrotnej kolejności.
Np. dla tablicy o elementach $11, 7, 8, 2, 5, 6, 1, 4, 10$ wypisane mają zostać wartości $10, 1, 5, 8, 11$.
Np. dla tablicy o elementach $3, 6, 8, 9, 2, 4, 1, 5$ wypisane mają zostać wartości $1, 2, 8, 3$.

b) Napisz program, który wypisze elementy tablicy w kolejności w jakiej w niej wystepują, tak aby elementy, których wartości się powtarzają były wypisane tylko za pierwszym razem pojawienia się w tablicy.
Np. dla tablicy o elementach $7, 4, 6, 7, 5, 5, 8, 6, 7$ zwrócone mają zostać wartości $7, 4, 6, 5, 8$.

c) Napisz program, który przesunie o 2 w lewo elementy tablicy tzn. element tab[0] będzie równy tab[2], element tab[1] równy tab[3], aż do tab[n-3], który będzie równy tab[n-1]. Ponadto tab[n-2] będzie równy tab[0] oraz tab[n-1] będzie równy tab[1].

d) Napisz funkcję typu bool, która sprawdza czy tablica czytana od przodu i od tyłu jest taka sama, to znaczy:
Np. dla tablicy o elementach $4, 6, 7, 8, 7, 6, 4$ zwrócona ma być wartość true;
Np. dla tablicy o elementach $5, 2, 9, 9, 2, 5$ zwrócona ma być wartość true;
Np. dla tablicy o elementach $1, 2, 3, 4, 5, 6$ zwrócona ma być wartość false;
Np. dla tablicy o elementach $1, 2, 3, 4, 2, 1$ zwrócona ma być wartość false;

Zadanie 6.(DODATKOWE +0,5 DO OCENY)
Na ostatniej stronie kolokwium przedstawiona jest implementacja algorytmu bąbelkowego sortowania w sposób niemalejący elementów typu int w tablicy. Mamy za zadanie posortować za jego pomocą tablicę złożoną z elementów 7 1 8 4 2 5 3 9 6 ale jak wiadomo NIE MA NIC ZA DARMO !
Wiadomo, że każdorazowe porównanie dwóch elementów kosztuje nas 2. (nawet gdy ich nie przestawiamy)
Natomiast, gdy okaże się że musimy zamienić ich kolejność to dochodzi dodatkowo koszt przestawienia 5.

a) Odpowiednio modyfikując algorytm, oblicz jaki będzie łączny koszt posortowania niemalejąco elementów podanej tablicy.
UWAGA. PAMIĘTAJ, ŻE KOLEJNE PODPUNKTY SĄ NIEZALEŻNE OD SIEBIE I NALEŻY UWZGLEDNIĆ TYLKO DODATKOWĄ INFORMACJĘ Z DANEGO PODPUNKTU, TJ. NIE NALEŻY ICH ŁACZYĆ.

b) Oblicz koszt posortowania niemalejąco, zakładając dodatkowo, że porównanie liczby parzystej z nieparzystą jest droższe i kosztuje nas 3 (zamiast 2), tj. porównanie np. 5 i 8 kosztuje 3, porównanie 6 i 4 kosztuje 2, porównanie 3 i 9 kosztuje 2 itd.

c) Oblicz koszt posortowania niemalejąco, zakładając dodatkowo, że przestawienie dwóch liczb parzystych jest tańsze i kosztuje tylko 4 (zamiast 5) tj. przestawienie 4 i 8 kosztuje 4, przestawienie 5 i 9 kosztuje 5, przestawienie 1 i 6 kosztuje 5 itd.

d) Co będzie nas kosztowało drożej, posortowanie w sposób niemalejący czy nierosnący podanej na początku tablicy ? Uzasadnić.

e) Dla jakiej tablicy 9-elementowej koszt posortowania będzie najmniejszy, a dla jakiej największy ? Ile bedą wynosić te koszty ?

SORTOWANIE BABELKOWE- ZAŁĄCZNIK DO ZADANIA 6

```
void sortujbabelekowo(unsigned int n, int * tab)
{
    int i,j,pom;
    for(i=0;i<n-1;i++)
    {
        for(j=0; j<n-1-i; j++)
        {
            if (tab[j]>tab[j+1])
            {
                pom=tab[j+1];
                tab[j+1]=tab[j];
                tab[j]=pom;
            }
        }
    }
}
```