

Zajęcia 3: Opracowanie prostych funkcji – cz. II

Do stworzonego na poprzednich zajęciach projektu (patrz instrukcja do zajęć 2) dodaj pięć nowych elementów a w każdym z nich zdefiniuj jedną z funkcji:

- (1) **void Odwroc(char* lancuch)**, która ma odwrócić ciąg znaków przekazanych w parametrze **lancuch**. Napis **lancuch** należy wczytać w funkcji głównej **main**, a następnie przekazać go do funkcji **Odwroc**, gdzie powinien zostać odwrócony bez tworzenia dodatkowych tablic. Po odwróceniu ciągu, wynik należy wypisać na ekran. Uwaga: w ciele funkcji nie należy używać funkcji z biblioteki **string.h**. Ponadto w ciele funkcji **Odwroc** nie można używać operacji wejścia/wyjścia.
- (2) **void ZamienLiter(char* lancuch)**, która ma zamienić litery napisu **lancuch** z dużych na małe i odwrotnie. Napis **lancuch** należy wczytać w funkcji głównej **main**, a następnie przekazać go do funkcji **ZamienLiter**, która ma zrealizować zamianę liter. Zmieniony napis należy wypisać na ekran. Uwaga: w ciele funkcji nie należy używać funkcji z biblioteki **string.h** oraz **ctype.h**. Ponadto w ciele funkcji **ZamienLiter** nie można używać operacji wejścia/wyjścia.
- (3) **void Podzielniki(int liczba)**, która ma wypisywać na ekran wszystkie podzielniki przesłanego do funkcji argumentu **liczba** w przypadku, gdy jest on większy od 0.
- (4) **int Zamien(int l, int p, int zapis[])**, która ma przedstawić liczbę **l** w systemie o podstawie **p**, a poszczególne cyfry (od najmniej znaczącej) umieścić w tablicy **zapis**.

Wskazówki do funkcji głównej:

- zadeklaruj tablicę **int zapis[20]** i przekaz ją do funkcji;
- wartość zwracaną przez funkcję **Zamien** przypisz do zmiennej pomocniczej;
- korzystając ze zmiennej pomocniczej, wypisz na ekran elementy tablicy **zapis** w odpowiedniej kolejności;
- wartość podstawy **p** powinna należeć do liczb ze zbioru **{2, 3, ..., 9}**.

Kroki w funkcji **Zamien**:

1. dopóki **l** jest różne od 0 wykonuj operację: **zapis[i]=l%p** gdzie **i** to zmienna iteracyjna;
2. po każdej iteracji zredukuj **l** wykonując **l = l / p**;
3. po zakończeniu pętli zwróć wartość zmiennej **i**.

W ciele funkcji **Zamien** nie można używać operacji wejścia/wyjścia.

- (5) **void WypiszKwadratyCyfr(int liczba)**, która ma wypisać na ekran kwadraty kolejnych cyfr liczby przekazanej jako argument.

Przykład: jeżeli **liczba = 7435**, na ekranie powinno się pojawić: **49 16 9 25**.

Wskazówka 1: w celu uzyskania kolejnych cyfr (od najbardziej znaczącej), należy najpierw wyznaczyć wartość wykładnika **n** na podstawie liczby cyfr argumentu **liczba**; np. – dla liczby czterocyfrowej, wykładnik **n = 3**. Następnie należy użyć operacji dzielenia całkowitego.

cyfra = liczba/10ⁿ = 7

W kolejnych krokach należy odpowiednio zmniejszać zarówno zmienną **liczba** jak i wykładnik **n**, aż do przetworzenia wszystkich cyfr.

Wskazówka 2: Aby wyznaczyć liczbę cyfr zmiennej **liczba** (a tym samym wartość wykładnika **n**), należy w pętli dzielić tę zmienną całkowicie przez 10, dopóki jej wartość jest większa niż 0. Liczba wykonanych iteracji (dzielenia) odpowiada liczbie cyfr liczby.

W pliku, w którym znajduje się definicja funkcji głównej **main** konieczne jest umieszczenie deklaracji zapowiadających wszystkich pięciu funkcji. Wywołanie w/w funkcji należy uwzględnić w instrukcji wielowariantowego wyboru **switch-case**.

Uwaga 1: tablica **elementy** ma być stworzona w sposób dynamiczny za pomocą wskaźnika, któremu należy przydzielić pamięć. W tym celu należy skorzystać z funkcji **malloc**.

Uwaga 2: tablica **lancuch** ma być stworzona w sposób dynamiczny za pomocą wskaźnika. W tym celu należy stworzyć dodatkową tablicę o stałej długości (np. **char tablica[20]**), wczytać do niej łańcuch z klawiatury, na jej podstawie przydzielić pamięć wskaźnikowi **lancuch** a następnie przekopiować jej zawartość do wskaźnika.