

Dokumentacja projektu		SO2
Autor	Michał Kula, 111105	Data oddania
Kierunek, rok	Informatyka, III rok, st. stacjonarne (3,5-1)	11.01.2023
Specjalizacja	Aplikacje Internetowe	
Grupa	LAB 3	Ocena
Temat projektu	<i>Mnożenie pisemne</i>	

1. Tematyka projektu:

Tematem projektu jest stworzenie programu w języku C który będzie realizował algorytm mnożenia pisemnego. Ponadto, interfejs programu został stworzony w oparciu o szablon zawarty na stronie <http://www.balois.pl/sys>

2. Wykorzystane technologie:

Język C w wersji 99

3. Wykorzystane narzędzia:

CLion – v. 2022.2.3

Kompilator GCC v. 6.3.0

4. Instrukcja uruchomienia programu:

Program można uruchomić poprzez IDE, lub wykonując polecenie gcc -o 'nazwa programu' mnozenie. Następnie należy uruchomić utworzony plik. Dane do programu są podawane po jego uruchomieniu.

5. Omówienie kodu

W kodzie zostały zawarte następujące zmienne:

a, b – do wczytania liczb

c – tablica trójwymiarowa zawierająca skopiowane liczby w celu łatwiejszej operacji na danych, przeniesienia, składniki dodawań oraz wynik. Tablica trójwymiarowa została użyta, ponieważ ilość dodawań jest dynamiczna tj. zależy od liczby cyfr liczby b

breakPoint – używany do walidacji poprawności danych

digitA, digitB – służą do pobierania cyfr z tablic a, b

result – wynik mnożenia digitA * digitB

writeLeftZeros - służą do ominięcia zer znajdujących się na początku liczb

x – używane do zwiększania o jeden rząd wielkości podczas obliczania końcowego wyniku.

i, j – używane w iteracjach pętli

Program rozpoczyna się od podania dwóch zmiennych a, b które są walidowane poprzez sprawdzenie czy każdy ze znaków jest cyfrą przy użyciu funkcji isdigit(). Jeżeli nie, pętla sprawdzająca jest przerywana i użytkownik proszony jest o podanie poprawnej liczby.

```
printf( format: "Mnozenie.c Zalozenie: a, b > 0\n");
printf( format: "Podaj a:");
int breakpoint; //do walidacji
do{
    breakpoint = 1;
    fgets( Buf: a, MaxCount: sizeof (a), File: stdin);
    for(int i = 0; a[i] != '\n'; i++){
        if(!isdigit( C: a[i])){
            printf( format: "Podaj poprawna liczbe.\n");
            breakpoint = 0;
            break;
        }
    }
} while (breakPoint == 0);
```

```
Podaj a:432#zxcz2
Podaj poprawna liczbe.
```

```
asd31
Podaj poprawna liczbe.
21
Podaj b:
```

3

Ze stringów a, b usuwany jest znak przejścia do nowej linii przy użyciu funkcji strcspn(), ponieważ powodowało to problemy podczas prawidłowego odczytania zmiennych.

```
//Usuwanie znaku /n
a[strcspn( Str: a, Control: "\n")] = 0;
```

Następnie tworzona jest tablica trójwymiarowa c, wypełniona zerami:

```

int c[strlen( Str: b)][strlen( Str: a) + strlen( Str: b) + 1][3];

//Wypełnienie c zerami
for (int i = 0; i < sizeof(c) / sizeof(c[0]); i++) {
    for (int j = 0; j < sizeof(c[0]) / sizeof(c[0][0]); j++) {
        for (int k = 0; k < sizeof(c[0][0]) / sizeof(int); k++) {
            c[i][j][k] = 0;
        }
    }
}

```

1 wymiar (numer wiersza) o rozmiarze strlen(b), ponieważ liczba iloczynów w mnożeniu pisemnym jest równa liczbie cyfr w liczbie b.

2 wymiar (numer kolumny) o rozmiarze strlen(a) + strlen(b) + 1, ponieważ maksymalna liczba cyfr w wyniku mnożenia jest o jeden większa od sumy liczb cyfr składników.

3 wymiar o stałym rozmiarze 3, w poszczególnych wymiarach umieszczane są przeniesienia, liczby a, b w celu łatwiejszego poruszania się w pętli i wyświetlania ich.

Po wypełnieniu tablicy c następuje główna część projektu. Wykonywany jest algorytm mnożenia pisemnego. Pobierane są kolejne cyfry z a, b i są mnożone przez siebie. Wynik tego mnożenia jest zapisywany w 3 zmiennych:

```

95 c[strlen( Str: b) - i - 1][i + j + 1][0] = (result + c[strlen( Str: b) - i - 1][i + j + 2][1]) / 10;
96 c[strlen( Str: b) - i - 1][i + j + 1][1] = (result + c[strlen( Str: b) - i - 1][i + j + 2][1]) / 10;
97 c[strlen( Str: b) - i - 1][i + j + 2][0] = (result + c[strlen( Str: b) - i - 1][i + j + 2][1]) % 10;

```

W komórce z linii 95 zapisywana jest pierwsza cyfra z sumy wyniku i ewentualnego przeniesienia z poprzedniego działania. Dla przykładu $8 * 7 = 56$, 6 zostanie wpisane później, natomiast 5 zostaje "zapamiętane" w tablicy dla przyszłych dodawań.

W komórce z linii 96 zapisywana jest ta sama cyfra, w celu późniejszego wykorzystania do wyświetlenia, tak jak w poprzednim przykładzie będzie to cyfra 5.

W linii 97 zapisywana jest reszta z dzielenia sumy wyniku i ewentualnego przeniesienia, dla przykładu w pierwszym kroku bierzemy wynik mnożenia czyli 56, do tego dodajemy ewentualne przeniesienie.

W ostatnim kroku następuje wypisanie wyników, zgodnie z interfejsem znajdującym się na wspomnianej wcześniej stronie:

1) Wypisane przeniesienia

```

printf( format: " ");
//Wypisywanie przeniesien
for (int i = strlen( Str: b) - 1; i >= 0; i--) {
    printf( format: "");
    for (int j = 0; j < strlen( Str: a) + strlen( Str: b) + 1; j++) {
        if (c[i][j][1] > 0)
            printf( format: "%d", c[i][j][1]);
        else
            printf( format: " ");
    }
    printf( format: "\n ");
}
}

```

2) Wypisanie zmiennych a, b

```

//Wypisywanie a
int writeLeftZeros = 1;
for (int i = 0; i < sizeof(c[0]) / sizeof(c[0][0]); i++) {
    if (c[0][i][2] > 0)
        writeLeftZeros = 0;
    if (writeLeftZeros == 0)
        printf( format: "%d", c[0][i][2]);
    else
        printf( format: " ");
}
printf( format: "\n");

//Wypisywanie b
writeLeftZeros = 1;
printf( format: "*");
for (int i = 0; i < sizeof(c[0]) / sizeof(c[0][0]); i++) {
    if (strlen( Str: b) == 1) {
        if(i == sizeof(c[0]) / sizeof(c[0][0]) - 1 )
            printf( format: "%c", b[0]);
        else
            printf( format: " ");
    } else {
        if (c[1][i][2] > 0)
            writeLeftZeros = 0;
        if (writeLeftZeros == 0) {
            printf( format: "%d", c[1][i][2]);
        } else
            printf( format: " ");
    }
}
}

```

3) Wypisanie kreski

```
for (int i = 0; i < sizeof(c[0]) / sizeof(c[0][0]); i++)  
    printf( format: "-");
```

4) Wypisanie poszczególnych składników dodawania

```
//Wypisanie dodawan  
printf( format: "\n ");  
for (int i = 0; i < sizeof(c) / sizeof(c[0]); i++) {  
    writeLeftZeros = 1;  
    for (int j = 0; j < sizeof(c[0]) / sizeof(c[0][0]); j++) {  
        if (c[i][j][0] > 0)  
            writeLeftZeros = 0;  
        if (writeLeftZeros == 0) {  
            printf( format: "%d", c[i][j][0]);  
        } else  
            printf( format: " ");  
    }  
  
    if (i == 0 && strlen( Str: b) > 1)  
        printf( format: "\n+");  
    else  
        printf( format: "\n");  
}
```

5) Wypisanie kolejnej kreski

```
if (strlen( Str: b) > 1) {  
    for (int i = 0; i < sizeof(c[0]) / sizeof(c[0][0]) + 1; i++)  
        printf( format: "-");  
}
```

6) Wypisanie końcowego wyniku

```
printf( format: "\n ");  
int result = 0;  
int x = 1;  
for (int i = sizeof(c[0]) / sizeof(c[0][0]) - 1; i >= 0; i--) {  
    result += (c[0][i][0] * x) + (c[1][i][0] * x);  
    x *= 10;  
}  
printf( format: "%d\n", result);  
}
```

Pełny przebieg programu:

```
Mnozenie.c Zalozenie: a, b > 0
Podaj a:372
Podaj b:57
a = 372, strlen(a) = 3, sizeof(a) = 20
b = 57, strlen(b) = 2, sizeof(b) = 20

    131
    251
    372
*   57
-----
    2604
+ 18600
-----
    21204
```

Przykład przy niepoprawnym formacie danych (np. znaki specjalne, litery):

```
Podaj a:432#zxcz2
Podaj poprawna liczbe.
```

```
asd31
Podaj poprawna liczbe.
21
Podaj b:
```