

Zadanie 1

Napisz funkcję

```
void rotate(int arr[], size_t n, size_t k)
```

przesuwającą n -elementową tablicę `arr` cyklicznie o k pozycji w lewo. Funkcja powinna modyfikować tablicę `arr` w miejscu oraz nie przyjmować żadnych założeń co do wzajemnej relacji zmiennych n oraz k . Upewnij siebie (i sprawdzającego) o poprawności swojej funkcji dostarczając testy jednostkowe. Oto przykładowy test:

```
int arr_1[] = {0,1,2,3};
rotate(arr_1, 4, 1);
assert(arr_1[0] == 1 && arr_1[1] == 2 && arr_1[2] == 3 && arr_1[3] == 0);
// taki sam wynik dla rotacji o 5 i 9
```

Pamiętaj, że należy również przetestować przypadki brzegowe, do których należy np. przesunięcie o 0 pozycji w lewo i przesuwanie tablicy jednoelementowej.

Zadanie 2

Liczby Stirlinga dane są wzorem rekurencyjnym

$$S(n,k) = \begin{cases} k * S(n-1, k) + S(n-1, k-1), & \text{dla } n \geq k \\ 0, & \text{dla } n < k \end{cases}$$

oraz warunkami brzegowymi $S(n,1) = 1, S(n,n) = 1, S(n,0) = 0, S(0,0) = 1$;

1. Napisz funkcję o sygnaturze

```
unsigned long Stirling(unsigned short n, unsigned short k)
```

rekurencyjnie obliczającą wartość $S(n, k)$.

2. Funkcja z poprzedniego podpunktu może być uruchamiana wielokrotnie podczas pracy programu. W tym wypadku wiele obliczeń będzie niepotrzebnie powtarzanych. By temu zaradzić, napisz rekurencyjną funkcję, która oblicza wartość $S(n,k)$ używając *memoryzacji*. Wyniki wywołań rekurencyjnych zapamiętuj w tablicy struktur o sygnaturze:

```
struct mem_entry {
    unsigned long value;
    bool valid;
};
```

Tablicę zadeklaruj globalnie tak:

```
struct mem_entry memory[N_MAX][N_MAX];
```

gdzie `N_MAX` jest stałą, którą zdefiniujesz dyrektywą `define`, np. `#define N_MAX 777`

3. Dostarcz testy jednostkowe rzetelnie testujące zaimplementowane funkcje, wykorzystując np. poniższą tabelkę wartości $S(n, k)$

n/k	0	1	2	3	4	5	6	7	8	9
0	1									
1	0	1								
2	0	1	1							
3	0	1	3	1						
4	0	1	7	6	1					
5	0	1	15	25	10	1				
6	0	1	31	90	65	15	1			
7	0	1	63	301	350	140	21	1		
8	0	1	127	966	1701	1050	266	28	1	
9	0	1	255	3025	7770	6951	2646	462	36	1

Zadanie 3

Zadanie sprawdzane automatycznie. Pojawi się w serwisie SKOS.