

## Podstawy Programowania 1 - kolokwium II, 19.01.2023, termin 0, gr. I02

We wszystkich zadaniach należy używać zapisu wskaźnikowego. Zapis tablicowy (użycie znaków `[]`) możliwy jest tylko przy inicjalizacji tablicy.

### Zadanie 1. (2 pkt.)

Napisz funkcję, która dla zadanego obrazu wykonuje odbicie górnego lewego trójkąta powstałego po przecięciu się dwóch przekątnych na prawy trójkąt. Dodatkowo w momencie wykonywania odbicia wykonuje negatyw wartości zadanego piksela. Wywołaj napisaną przez siebie funkcję na tablicy `photo`.

Obraz wejściowy:



Ilustracja, które fragmenty obrazu należy skopiować i gdzie umieścić. Należy skopiować biały trójkąt na czarny :



Obraz po prawidłowych przekształceniach:



Wykorzystaj template kodu:

[secdev.pl/code\\_template.c](http://secdev.pl/code_template.c)

Oraz Kompilator:

[secdev.pl/compile.php](http://secdev.pl/compile.php)

lub

[kolokwium.secdev.pl/code\\_template.c](http://kolokwium.secdev.pl/code_template.c)

Oraz Kompilator:

[kolokwium.secdev.pl/compile.php](http://kolokwium.secdev.pl/compile.php)

## Zadanie 2. (3 pkt.)

Napisz funkcje:

```
int read_numbers(int *numbers, int numbers_size, int stop_value);
void display_numbers(const int *numbers, int numbers_size);
void sort_by_circles(int *numbers, int numbers_size);
```

Funkcja `read_numbers` wczytuje od użytkownika tablicę liczb naturalnych. Oprócz tablicy `numbers` i jej maksymalnego rozmiaru `numbers_size`, ma przyjąć wartość elementu `stop_value`, po wczytaniu którego funkcja kończy działanie i zwraca odpowiednią wartość. Wartością zwracaną przez funkcję `read_numbers` ma być liczba wczytanych elementów.

- W przypadku przekazania do funkcji błędnych danych (np. tablicy o pojemności 0), funkcja powinna zwrócić wartość -1
- W przypadku wprowadzenia przez użytkownika wartości innych niż liczbowe, funkcja powinna zwrócić wartość -2.
- W przypadku wprowadzenia przez użytkownika wartości ujemnych funkcja powinna zwrócić wartość -3.

Funkcja `display_numbers` jako parametry przyjmuje tablicę do wyświetlenia oraz jej długość. Funkcje inne niż `display_numbers` nie mogą wyświetlać żadnych wartości.

Funkcja `sort_by_circles` ma sortować wejściową tablicę w kolejności od liczb, które mają najmniejszą ilość kółeczek, do tych które mają największą ilość kółeczek. Jeżeli liczby mają taką samą liczbę kółeczek, to pierwsze mają być liczby mniejsze, a następnie liczby większe. Dla przykładu:

- liczba 0 ma 1 kółeczko
- liczba 1 ma 0 kółeczek
- liczba 2 ma 0 kółeczek
- liczba 3 ma 0 kółeczek
- liczba 4 ma 0 kółeczek
- liczba 5 ma 0 kółeczek
- liczba 6 ma 1 kółeczko
- liczba 7 ma 0 kółeczek
- liczba 8 ma 2 kółeczka
- liczba 9 ma 1 kółeczko
- liczba 10 ma 1 kółeczko (0 kółeczek z jedynki i 1 kółeczko z zera)
- ...
- liczba 286 ma 3 kółeczka (0+2+1)

Napisz program który pobierze od użytkownika liczby i zapisze je do tablicy wykorzystując funkcję `read_numbers` i wyświetli pobrane liczby za pomocą funkcji `display_numbers`. Wartość `stop_value` ustaw na 404. Następnie program ma posortować tablicę z wykorzystaniem funkcji `sort_by_circles` oraz wyświetlić posortowaną tablicę.

Przykładowe działanie programu:

```
podaj liczby: 10 34 2 98 268 36 404
10 34 2 98 268 36
2 34 10 36 98 268
```

### Zadanie 3. (4 pkt.)

Podziel obraz na ramki o szerokości 10 x 10 pikseli i nadaj im numery od 1 do 625. Numerację zacznij od lewego górnego rogu obrazu i skończ na prawym dolnym: Przykład numeracji poniżej:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625

Napisz funkcję rekurencyjną:

```
int sequence_recursive(int n);
```

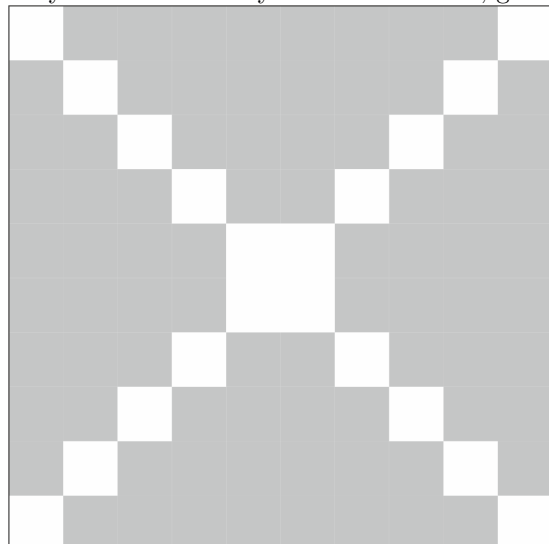
Parametrem wejściowym funkcji jest liczba naturalna  $n$ .

Funkcja ma zwracać  $n$ -ty elementu ciągu opisanego wzorem:

$$\begin{cases} a_0 = 2 \\ a_1 = 1 \\ a_n = a_{n-1} + a_{n-2} \end{cases}$$

Wykorzystując wyżej opisaną funkcję należy wygenerować początkowe wartości ciągu i zapisać je do tablicy. Wartości należy wyznaczać tak długo, aż zostanie wyznaczona wartość większa od 625 (nr. ostatniej ramki). Na obrazie należy oznaczyć znakiem X ramki o numerach odpowiadających wyznaczonym wartościom ciągu. Znak X należy narysować zamalowując na biały kolor (wartość 255) piksele leżące na obu przekątnych ramki.

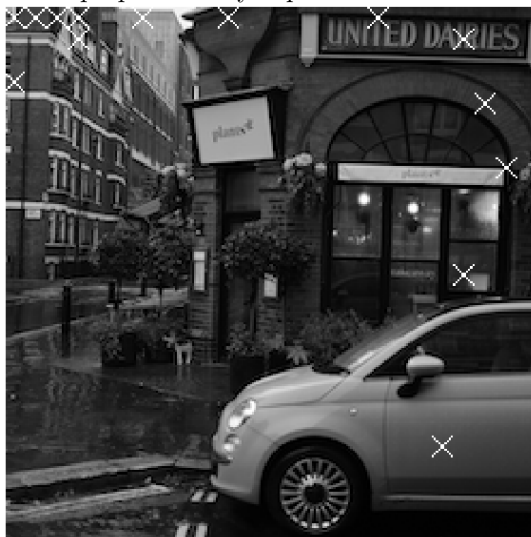
Przykład ramki z narysowanymi znakami X, gdzie każdy kwadrat to jeden piksel.



Obraz wejściowy:



Obraz po prawidłowych przekształceniach:



*Przeliczenie punktów na ocenę:*

- 0 - 4.5 punktu → ocena 2
- 5 - 5.5 punktu → ocena 3
- 6 - 6.5 punktu → ocena 3,5
- 7 - 7.5 punktu → ocena 4
- 8 - 8.5 punktu → ocena 4,5
- 9 punktów - ocena 5