



WSTĘP DO INFORMATYKI

KOLOKWIMUM

18-01-2008

Zadania

Zadanie 1 (15 pkt.)

Prześledzić działanie następującego programu

```
#include<stdio.h> /* 1*/
int funny(int x, int t[]) { /* 2*/
    int *w = &x; /* 3*/
    *w *= x/2; /* 4*/
    *(t+x) = 0; /* 5*/
    return *w+1; /* 6*/
} /* 7*/
int main(void) { /* 8*/
    int x, *w, **ww; /* 9*/
    int t[3] = {1, 2, 3}; /*10*/
    w = &x; /*11*/
    ww = &w; /*12*/
    for(x=0; x<4; x*=2) { /*13*/
        *w += funny(**ww, t); /*14*/
    } /*15*/
    return 0; /*16*/
} /*17*/
```

Zadanie 2 (15 pkt.)

Napisać program, który wyzeruje elementy tuż nad i tuż pod diagonalą macierzy jeśli element tejże diagonali jest ujemny. Założyć, że macierz jest wymiaru 10×10 , zawiera liczby zmiennoprzecinkowe podwójnej precyzji. Program powinien najpierw zapytać użytkownika o każdy z elementów macierzy, a po jej modyfikacji wypisać ją na ekran (wierszami).

Przykładowa modyfikacja (elementy wyzerowane zostały zaznaczone):

$$\begin{bmatrix} 1 & 2 & 3 & -4 \\ 2 & -2 & -1 & 0 \\ -1 & -2 & 1 & 3 \\ 0 & 2 & -1 & -2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & \boxed{0} & 3 & -4 \\ 2 & -2 & -1 & 0 \\ -1 & \boxed{0} & 1 & \boxed{0} \\ 0 & 2 & -1 & -2 \end{bmatrix}$$

Zadanie 3 (10 pkt.)

Prześledzić działanie następującego programu

```
#include<stdio.h> /* 1*/
#include<stdlib.h> /* 2*/
void main(void) { /* 3*/
    struct s { /* 4*/
        int s; /* 5*/
        struct s *ss; /* 6*/
    } a = {1, NULL}; /* 7*/
    a.ss = (struct s*) malloc(sizeof(struct s)); /* 8*/
    a.ss->s = 2; /* 9*/
    (*(a.ss)).ss = (struct s*) malloc(sizeof(struct s)); /*10*/
    a.ss->ss->s = 3; /*11*/
    a.ss->ss->ss = NULL; /*12*/
    printf("%d %d %d\n", a.s, a.ss->s, a.ss->ss->s); /*13*/
} /*14*/
```



Rozwiązania

Rozwiązanie zadania 1

/* 9 */		
60	?	x
61	?	w
62	?	ww
63	?	ww
64		
65		
66		
67		
68		

/* 10 */		
60	?	x
61	?	w
62	?	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 11 */		
60	?	x
61	60	w
62	?	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 12 */		
60	?	x
61	60	w
62	61	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 13 */		
60	0	x
61	60	w
62	61	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 2 */		
60	0	x
61	60	w
62	61	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66	0	x'
67	63	t'
68		

/* 3 */		
60	0	x
61	60	w
62	61	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66	0	x'
67	63	t'
68	66	w'

/* 4 */		
60	0	x
61	60	w
62	61	ww
63	1	t[0]
64	2	t[1]
65	3	t[2]
66	0	x'
67	63	t'
68	66	w'

/* 5 */		
60	0	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66	0	x'
67	63	t'
68	66	w'

war. true		
/* 6 */		
60	0	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66	0	x'
67	63	t'
68	66	w'

/* 14 */		
60	1	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 13 */		
60	2	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 2 */		
60	2	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66	2	x'
67	63	t'
68		

/* 3 */		
60	2	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66	2	x'
67	63	t'
68	60	w'

funny zwraca 1		
/* 4 */		
60	2	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66	2	x'
67	63	t'
68	60	w'

/* 5 */		
60	2	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	0	t[2]
66	2	x'
67	63	t'
68	60	w'

war. true		
/* 6 */		
60	2	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66	2	x'
67	63	t'
68	60	w'

/* 14 */		
60	5	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 13 */		
60	10	x
61	60	w
62	61	ww
63	0	t[0]
64	2	t[1]
65	3	t[2]
66		
67		
68		

/* 16 */		
60		
61		
62		
63		
64		
65		
66		
67		
68		

funny zwraca 3

war. false

main zwraca 0

Rozwiązanie zadania 2

```
#include <stdio.h>

#define R 3
int main(void) {
    int i, j;
    float macierz[R][R];
    int x;
    printf("Program zmieniający macierze\n");
```



```

/* Pętla wczytująca macierz */
for(i=0; i<R; i++) {
    for(j=0; j<R; j++) {
        printf("Podaj (%d,%d) element macierzy: ",i+1, j+1);
        scanf("%f", &macierz[i][j]);
    }
}
/* Pętla modyfikująca macierz */
/* Iterujemy po elementach diagonalni */
for(i=0; i<R; i++) {
    /* Czy i-ty element diagonalni jest ujemny? */
    if(macierz[i][i] < 0) {
        /* Zabezpieczenie na wypadek pierwszego elementu diagonalni (nie ma nic nad) */
        if(i>0) {
            macierz[i-1][i] = 0;
        }
        /* Zabezpieczenie na wypadek ostatniego elementu diagonalni (nie ma nic pod) */
        if(i<R-1) {
            macierz[i+1][i] = 0;
        }
    }
}
/* Wypisanie przetworzonej macierzy na ekran */
printf("Przetworzona macierz:\n");
for(i=0; i<R; i++) {
    for(j=0; j<R; j++) {
        printf("%f ",macierz[i][j]);
    }
    printf("\n");
}
return 0;
}

```

Rozwiązanie zadania 3

/* 7 */			/* 8 */			/* 9 */			/* 10 */			/* 11 */		
60	1	a.s	60	1	a.s	60	1	a.s	60	1	a.s	60	1	a.s
61	NULL	a.ss	61	101	a.ss	61	101	a.ss	61	101	a.ss	61	101	a.ss
	:			:			:			:			:	
			101	?	.s	101	2	.s	101	2	.s	101	2	.s
			102	?	.ss	102	?	.ss	102	201	.ss	102	201	.ss
										:			:	
									201	?	.s	201	3	.s
									202	?	.ss	202	?	.ss

/* 12 */			/* 13 */			/* 14 */		
60	1	a.s	60	1	a.s	60		
61	101	a.ss	61	101	a.ss	61		
	:			:			:	
101	2	.s	101	2	.s	101	2	.s
102	201	.ss	102	201	.ss	102	201	.ss
	:			:			:	
201	3	.s	201	3	.s	201	3	.s
202	NULL	.ss	202	NULL	.ss	202	NULL	.ss

EKRAN:1 2 3