

PC - testul 2

1. (5p) Scrieți în grila alăturată, în ordine, cele 10 valori care se afișează la ieșirea standard, pe rânduri separate, în urma execuției programului C de mai jos. Presupuneți că alocarea dinamică de memorie care apare este întotdeauna realizată cu succes.

<pre>#include <stdio.h> #include <stdlib.h> #define TEST int x; #undef TEST #ifndef TEST double d = 4.56; #else double d = 7.89; #endif #define DUBLU(z) 2*z float dublu(int z) { z *= 2; return z; } void o_suta() { x = 100; } int seteaza(int *p, int *val) { static int nr=0; *p=*val; nr++; return nr; } void zero(void *x, int n) { short int *t = x; for (int i=0; i<n; i++) t[i]=0; }</pre>	<pre>int suma(int x[], int n) { int s=0; for (int i=0; i<n; i++) s += *(x+i); return s; } int main() { printf("%f\n", d); printf("%d\n", x); { int x = 50; o_suta(); printf("%d\n", x); } printf("%d\n", x); printf("%d\n", DUBLU(9-6)); float f = 3.6f; float r = dublu(f); printf("%f\n", f+r); int *t = (int*)malloc(10*sizeof(int)); printf("%s\n", sizeof(t)>9 ? "MARE":"MIC"); int rez; for (int i=0; i<10; i++) rez=seteaza(t+i,&i); printf("%d\n", rez+ *t + t[9]); int a[10]={0,1,2,3,4,5,6,7,8,9}; printf("%d\n", t!=a); zero(a, 8); printf("%d\n", suma(a, 10)); return 0; }</pre>	<p>Rezultate afișate (câte unul pe linie):</p> <table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>										

2. (4p) Scrieți funcțiile *genereaza* și respectiv *afiseaza* care lipsesc din următorul program C astfel încât acesta să genereze și respectiv să afișeze la ieșirea standard o matrice pătratică care să aibă pe liniile sale permutările circulare ale unui șir de valori cu numere întregi dat. Programul afișează în final și numărul total de elemente din matricea creată. Nu este permisă utilizarea de variabile globale! Nu se va face risipă de memorie! Se presupune că toate datele de intrare sunt valide.

<pre>#include <stdio.h> #include <stdlib.h> int main() { int a[] = {10,20,30,40}; int n = sizeof(a)/sizeof(*a); int **mat; int nr; genereaza(a,n,&mat,&nr); afiseaza(mat,n); printf("Nr. total de elemente: %d\n",nr); return 0; }</pre>

Rezultate afișate la ieșirea standard:

```
10 20 30 40
20 30 40 10
30 40 10 20
40 10 20 30
Nr. total de elemente: 16
```