

Indicații

- Testul conține 3 subiecte, durează 120 de minute și valorează 50 de puncte, cu alte 5 bonus.
- Se pot obține punctaje parțiale pentru un subiect numai dacă acest lucru este specificat.
- Pentru a fi punctată, o rezolvare **trebuie** să includă și metoda de verificare a funcționalității acesteia.
- Pentru subiectul 1, pentru afișarea la ecran este recomandată utilizarea macroului `PRINTF32`.
- Pentru subiectul 2, pentru afișarea la ecran este obligatorie utilizarea funcției de bibliotecă `printf()`.
- Ordinea de rezolvare a subiectelor este la alegerea voastră.
- Subiectele se rezolvă pe mașina virtuală de PCLP2.

Subiecte

Subiectul 1

Toate subpunctele de mai jos se vor rezolva în fișierul `main.asm`.

- [5p] Copiați vectorul definit în zona de date `.rodata`, `arr`, pe stivă. Afișați vectorul de pe stivă.
- [5p] Dorim ca fiecare element să fie par. Pentru a realiza acest lucru, **incrementați toate valorile impare** din vectorul realizat la subpunctul a. Afișați vectorul de pe stivă.
- [5p] Pentru fiecare poziție `i` din vectorul `arr`, dorim să aflăm dacă suma tuturor elementelor din `arr` fără cel de pe poziția `i` încapă într-o variabilă de tip `unsigned short`.

Creați un nou vector `bools` cu `N` elemente în zona de date `.data` care va conține 1 pe pozițiile care îndeplinesc proprietatea cerută, sau 0 în caz contrar. Afișați acest vector.

- [5p] Afișați elementele din `arr2` care se pot scrie utilizând cel mult `i` biți, unde `i` este poziția elementului în `arr2`. (**Hint:** Indexarea vectorului începe de la 0)

Subiectul 2

ATENȚIE: Pentru subpunctele de mai jos **nu veți putea folosi** macro-ul `PRINTF32`!

Toate subpunctele de mai jos se vor rezolva în fișierul `main.asm`.

- [5p] Alocați pe stivă în main un șir de caractere de `N` octeți. Copiați conținutul din `rostring`, aflat în `.rodata`, în acest șir de caractere, utilizând funcția `strcpy`, apoi afișați noul șir pentru a vă verifica.
- [5p] Implementați funcția `void replace_chars(char *string, char new, char *old)` care înlocuiește fiecare apariție a caracterelor din `string`-ul `old` cu caracterul `new` în `string`. (**Hint:** Puteți folosi funcții de bibliotecă declarate în `string.h`)

Apelați funcția în main folosind ca argument șirul de caractere alocat la subpunctul a pentru a înlocui caracterele `a`, `h` și `i` cu `b` și afișați șirul de caractere modificat.

- [5p] Implementați funcția `int count_words(char *string)` care numără câte cuvinte conține șirul de caractere `string`. Cuvintele vor fi separate de câte un singur spațiu.

Apelați funcția în main folosind ca argument șirul de caractere `rostring` și afișați rezultatul.

- [5p] Implementați funcția `char *random_key(int length)` care alocă pe heap exact `length + 1` octeți și salvează în această zonă un șir de `length` caractere format din litere mici random. Se va întoarce acest șir.

În main, apelați funcția folosind ca argument `KEY_LENGTH`, și afișați rezultatul pentru a vă verifica.

Subiectul 3

- [5p] Fără a modifica binarul `wired.o`, găsiți un mod de a apela `part_one` din `main.c` astfel încât să se afișeze mesajul `No matter where you are....`
- [5p] Fără a modifica binarul `wired.o`, găsiți un mod de a apela `part_two` din `main.c` astfel încât să se afișeze mesajul `... everyone is always connected..`
- [5p] Fără a modifica executabilul `hack`, rulați-l astfel încât să se afișeze mesajul `Welcome! You are root!.`