ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

TRANSPOZICE MATICE

Martin Hamet
A14B0254P

Zadání:

Nejprve se zadá parametr n, potom postupně po řádcích matice A(int16). Výstupem bude matice AT. Transpozice musí proběhnout v rozsahu paměti, ve které je uložena matice A. Paměť musí být dimenzována pro matice 7x7. Matice musí být v paměti uložena v souvislém prostoru (tj. za posledním prvkem určitého řádku následuje v paměti bezprostředně první prvek následujícího řádku).

Popis programu:

Program začíná na návěští _start. Program provede nezbytnou inicializaci a načtení parametru n z konzole. Dále pomocí podprogramů načte příslušnou matici z konzole, provede její transpozici a vypíše transponovanou matici.

Proměnné v programu:

Název	adresa v paměti	popis
prompt	FF4000	značka “>” pro konzoli
par_prompt	FF4004	ukazatel na značku “>”
par_out	FF4008	ukazatel na výstupní buffer
par_in	FF400C	ukazatel na vstupní buffer
out_buf	FF4010	vstupní buffer
in_buf	FF4074	výstupní buffer
PIN	FF40D8	pomocný ukazatel na vstupní buffer
POUT	FF40Dc	pomocný ukazatel na výstupní buffer
num_count	FF40E0	počítadlo cifer pro převod z desítkové soustavy
matice	FF40E4	prostor pro uložení matice
PM	FF41D9	pomocný ukazatel na matici
P_ROW	FF41DD	pomocný ukazatel na řádek matice
P_COL	FF41E1	pomocný ukazatel na sloupec matice
size	FF41E5	proměná obsahující řád matice
stck	FF424C	zásobník programu

Podprogramy:

init:

Vynuluje registry a čítač počtu cifer, nastaví pomocné ukazatele vstupního a výstupního bufferu a ukazatele na matici.

clr_reg:

Vynuluje používané registry kromě ER0, který se používá jako vstup/výstup některých podprogramů.

read_cmd:

Vypíše znak “>” do konzole, přečte zadanou řádku v konzoli a uloží její obsah do vstupního bufferu.

convert:

Provede převod čísla ve vstupním bufferu z desítkové do dvojkové soustavy a výsledné číslo ponechá v registu ER0. Dále příslušně nastaví ukazatele na vstupní buffer podle následujícího znaku (tj. mezera nebo konec řádku).

last:

Nastaví ukazatel vstupního bufferu na poslední cifru aktuálně převáděného čísla a nastaví čítač cifer.

insert_mat:

Vloží připravené číslo z ER0 na příslušnou pozici v matici.

read_line:

Opakovaným voláním procedur convert a insert_mat docílí převedení a uložení jedné řádky matice z konzole do paměti.

shift_row:

Posune pomocný ukazatel na řádky matice o jedno místo.

shift_col:

Posune pomocný ukazatel na sloupce matice o jedno místo.

shift_dia:

Posune pomocný ukazatel po diagonále v matici.

swap:

Vymění hodnoty dvou pozic v matici podle adres ER4 a ER5.

transform:

Pomocí procedur shift_* a swap provede transpozici matice.

controll:

Zjistí zda číslo v registru R0L je větší než 9 a pokud ano přičte k R0L hodnotu 0x37, jinak přičte pouze 0x30. (Pro převod na znak ASCII.)

raw_out_line:

Vypíše jeden řádek matice do konzole (tak jak je uložený v paměti v 16 soustavě). Jednotlivá čísla jsou oddělená mezerou.

raw_write_out:

Opakovaným voláním procedury raw_out_line vypíše celou matici do konzole.

Ovládání programu:

Program ihned po spuštění vypíše do konzole znak “>” a čeká na zadání řádu matice a potvrzení stiskem klávesy “enter”. Dále vypisuje vždy jeden znak “>” a očekává zadání řádku matice kde jsou jednotlivá čísla oddělena mezerou a celý řádek ukončený stisknutím “enter”. Po zadání všech řádků program Provede transpozici matice a výslednou matici vypíše v 16 soustavě do konzole.

Program nijak nekontroluje správnost zadaných hodnot nebo jejich počet v řádku a je tedy nutné zadávat správná vstupní data.

Obsah zásobníku při volání jednotlivých podprogramů:

Návratové adresy jsou zapsány tučně a vrchol zásobníku je myšlen jako poslední řádek. Pro jednoduchost byla do programu zadána matice 1. řádu s jednociferným číslem kvůli často opakovanému volání podprogramů. (Zásobník by měl podobný obsah, ale ve větším rozsahu.)

Odrážky u názvu podprogramu označují úroveň zanoření.

Podprogram	**Obsah StackPointeru**	**Obsah Zásobníku**
init	**0000 040A**	**0000 040A**
read_cmd	**0000 040E**	**0000 040E**
read_line	**0000 0422**	0101 **0000 0422**
- read_cmd	**0000 04B4**	0101 0000 0422

		0000 04B4
- convert	**0000 04C0**	0101 0000 0422 0101 **0000 04C0**
- - last	**0000 04D0**	0101 0000 0422 0000 04C0 **0000 04D0**
- - clr_reg	**0000 0530**	0101 0000 0422 0101 0000 04C0 **0000 0530**
- insert_mat	**0000 04c4**	0101 0000 0422 0101 **0000 04C4**
transfrorm	**0000 043A**	**0000 043A**
- shift_row	**0000 05BA**	0000 043A **0000 05BA**
- shift_col	**0000 05BE**	0000 043A **0000 05BE**
- swap	**0000 05C2**	0000 043A **0000 05C2**
- shift_dia	**0000 05D4**	0000 043A **0000 05D4**
- - shift_col	**0000 0634**	0000 043A 00FF41E4 00FF41E4 **0000 0634**
- - shift_row	**0000 063A**	0000 043A 00FF41E4 00FF41E4 **0000 063A**
clr_reg	**0000 043E**	**0000 043E**
raw_write_out	**0000 0442**	**0000 0442**

- raw_out_line	**0000 067E**	0000 0442 **0000 067E**
- - controll	**0000 06AA**	0000 0442 0000 067E **0000 06AA**
- - controll	**0000 06BA**	0000 0442 0000 067E **0000 06BA**
- - controll	**0000 06CC**	0000 0442 0000 067E **0000 06CC**
- - controll	**0000 06DC**	0000 0442 0000 067E **0000 06DC**