

*Лабораторијска вежба, траје 2 сата и 15 минута***1.**

а) (7) На x86 асемблеру написати потпрограм за обраду низа бројева чији је потпис:

```
void obrada(int niz[], int n);
```

Потпрограм сместити у датотеку **L:\z1a.s** и при имплементацији користити *stdcall* конвенцију позивања потпрограма. Потпрограм треба да изврши сортирање низа у нерастућем поретку.

б) (1) На x86 асемблеру написати део програма који у секцији за податке алоцира простор за низ од 100 елемената величине 4B. Алоцирани простор треба попунити понављањем вредности 1, 2, 3, 4, 5, 6 и 7. Овај део програма сместити у датотеку **L:\z1b.s**. Почетак алоцираног простора означити лабелом **NIZ_POSETAK**, док прву адресу после краја низа треба означити лабелом **NIZ_KRAJ**. Садржај датотеке треба да буде минималне могуће величине (не сме садржати друге секције осим секције за податке).

в) (2) Написати програм на програмском језику C који позива потпрограм **obrada** за низ алоциран у тачки б), при чему програм треба да користи лабеле описане тачком б) како би израчунао број елемената низа. После позива потпрограма **obrada**, програм треба да испише на стандардни излаз садржај низа (елементи низа у једном реду раздвојени блатком знаком). Програм сместити у датотеку **L:\z1c.c**. Комплетан изворни код програма мора бити смештен у наведене 3 датотеке (не смеју постојати друге датотеке).

2.

а) (8) На ARM асемблеру написати рекурзивни потпрограм за обраду једноструко уланчане листе чији је потпис:

```
elem_t obrada(elem_t *glava, int x);
```

Тип **elem_t**, као и код за учитавање једноструко уланчане листе дати су у датотеци **lista.c**. Потпрограм сместити у датотеку **L:\z2a.s**. Конвенција позивања потпрограма треба да буде у складу са *ARM Architecture Procedure Call Standard (AAPCS)*. Потпрограм треба да испише све елементе листе (вредности поља **podatak** у једном реду раздвојене блатком знаком) почев од последњег. Потпрограм враћа копију елемента листе за који важи да је апсолутна разлика вредности поља **podatak** и параметра **x** најмања. У случају било какве нерегуларне ситуације потпрограм враћа структуру чија оба поља имају вредност нула. На самом почетку потпрограма треба позвати потпрограм **test** дефинисан у датотеци **lista.c**.

б) (2) На ARM асемблеру написати програм (потпрограм **main**) који са стандардног улаза учита вредност за параметар **x** потпрограма **obrada** као и број елемената у листи, затим позивом потпрограма **ucitajListu** формира једноструко уланчану листу учитавањем задатог броја бројева, након чега позове потпрограм **obrada** и испише у новом реду вредност поља **podatak** елемента листе који је вратио потпрограм **obrada**, док на крају обрише креирану листу позивом потпрограма **oslobodi**. Програм сместити у датотеку **L:\z2b.s**.