

Pro všechny příklady uvažujte konvenci pro model flat32 (Linux) a jazyk C. Není-li uvedeno jinak, pište kód v JSI. Řetězce jsou ukončeny binární nulou. *Int* je 32-bitový znaménkový. *Char* je 8-bitový znaménkový. Ve funkcích není potřeba ukládat a obnovovat registry.

1. Napište funkci pro rotaci 64 bitového čísla vlevo o N bitů. (8)

```
long long rol64( long long cislo, int N ) { ...  
    return cislo <<<< N;  
}
```

2. Napište funkci pro sečtení všech prvků pole x délky N . Návrátová hodnota je 64-bitová. (6)

```
long long soucet( int *x, int N ) {  
    ...  
    return suma;  
}
```

3. Může v předchozím příkladě dojít k přetečení? Svůj názor zdůvodněte. (5)

4. Navrhněte funkci pro sčítání celých čísel se saturací. Saturace znamená, že v případech přetečení/podtečení, bude výsledná hodnota MAX/MIN rozsahu typu int. (10)

```
int satur_plus( int x, int y ) {  
    return saturovane_x_plus_y;  
}
```

5. Napište funkci pro násobení 64 bitového čísla 32 bitovým číslem. Přetečení neuvažujte. (8)

```
long long nasob( long long ll, int i ) {  
    return ll * i;  
}
```

6. Napište funkci, která odstraní z řetězce úvodní mezery. Návrátová hodnota je upravený řetězec. (8)

```
char *trim_space( char *source ) {  
    char *s = source;  
    char *d = source;  
    while ( *s != '\0' && *s == ' ' ) s++;  
    while ( ( *d = *s ) != '\0' ) { s++; d++; }  
    return source;  
}
```