

```
;
;--- Task 2 ---
;
; Create a function: void* task22(const char *pA, int N, char x) to search an array pA of N 8bit signed
; values for the first occurrence of the value x. The function returns pointer to the value in the array.
; The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.
;
; Function parameters:
;   pA: pointer to the array A to search in
;   N: length of the array A
;   x: value to be searched for
;
; Return values:
;   EAX = 0 if the pointer pA is invalid (pA == 0) or N <= 0 or the value x has not been found in the array
;   EAX = pointer to the value x in the array (the array elements are indexed from 0)
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

--- Task 2 ---

Create a function: `unsigned int task22(const unsigned int *pA, int N)` to find minimum in an array pA of N 32bit unsigned values. The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.

Function parameters:

- pA: pointer to the array A
- N: length of the array A

Return values:

- EAX = 0x80000000 if the pointer pA is invalid (`pA == 0`) or `N <= 0`
- EAX = value of the 32bit unsigned minimum

Important:

- the function MUST preserve content of all the registers except for the EAX and flags registers.

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const int *pA, int N), která v poli 32bitových čísel se znaménkem pA
; nalezne maximum a vrátí jeho hodnotu v registru EAX. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence PASCAL.
;
; Parametry funkce:
;   pA: ukazatel na začátek pole 32bitových prvků se znaménkem
;   N: počet prvků pole pA (32bitové číslo se znaménkem)
;
; Návratová hodnota:
;   EAX = 0x80000000, když je pA neplatný ukazatel (pA == 0) nebo N <= 0
;   EAX = hodnota 32bitového maxima se znaménkem.
;
; Důležité:
;   Funkce musí zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned char *pA, int N, unsigned char x) to count all occurrences of the values greater than x
; in an array pA of N 8bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array greater than x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const int *pA, int N, int x) to count all occurrences of the values equal to x
; in an array pA of N 32bit signed values.
; The parameters are passed, the stack is cleaned and the result is returned according to the STDCALL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array equal to x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: unsigned short task22(const unsigned short *pA, int N), která v poli 16bitových čísel bez znaménka pA
; nalezne maximum a vrátí jeho hodnotu v registru AX. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence STDCALL.
;
; Parametry funkce:
;   pA: ukazatel na začátek pole 16bitových prvků bez znaménka
;   N: počet prvků pole pA (32bitové číslo se znaménkem)
;
; Návrátová hodnota:
;   EAX = 0x80000000, když je pA neplatný ukazatel (pA == 0) nebo N <= 0
;   AX = hodnota 16bitového maxima bez znaménka.
;
; Důležité:
;   Funkce musí zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;--- Task 2 ---  
;  
; Create a function: int task22(const char *pA, int N, char x) to search an array pA of N 8bit signed  
; values for the first occurrence of the value x. The function returns position of the value in the array.  
; The parameters are passed, the stack is cleaned and the result is returned according to the PASCAL calling convention.  
;  
; Function parameters:  
;   pA: pointer to the array A to search in  
;   N: length of the array A  
;   x: value to be searched for  
;  
; Return values:  
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N <= 0 or the value x has not been found in the array  
;   EAX = position of the value x in the array (the array elements are indexed from 0)  
;  
; Important:  
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.  
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned char *pA, int N, unsigned char x) to count all occurrences of the values lower than x
; in an array pA of N 8bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the STDCALL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array lower than x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```



```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned short *pA, int N, unsigned short x) to count all occurrences of the values equal to x
; in an array pA of N 16bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array equal to x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned char *pA, int N, unsigned char x) to count all occurrences of the values lower than x
; in an array pA of N 8bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the STDCALL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array lower than x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const unsigned char *pA, int N, unsigned char x), která v poli 8bitových čísel bez znaménka pA
; o délce N nalezne první výskyt hodnoty x a vrátí index hledané hodnoty.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence PASCAL.
;
; Parametry funkce:
;   pA: ukazatel na pole A
;   N: počet prvků pole A
;   x: hodnota, kterou hledáme
;
; Návrátová hodnota:
;   EAX = -1, pokud je pA neplatný ukazatel (pA == 0) nebo pokud N <= 0 nebo pokud hodnota x nebyla v poli nalezena
;   EAX = index hledané hodnoty (prvky pole jsou indexovány od 0)
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: short task22(const short *pA, int N), která v poli 16bitových čísel se znaménkem pA
; nalezne minimum a vrátí jeho hodnotu v registru AX. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence PASCAL.
;
; Parametry funkce:
;   pA: ukazatel na začátek pole 16bitových prvků se znaménkem
;   N: počet prvků pole pA (32bitové číslo se znaménkem)
;
; Návratová hodnota:
;   EAX = 0x80000000, když je pA neplatný ukazatel (pA == 0) nebo N <= 0
;   AX = hodnota 16bitového minima se znaménkem.
;
; Důležité:
;   Funkce musí zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;--- Úkol 2 ---  
;  
; Naprogramujte funkci: int task22(const short *pA, int N, short x), která v poli 16bitových čísel se znaménkem pA  
; o délce N nalezne poslední výskyt hodnoty x a vrátí ukazatel na hledanou hodnotu.  
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.  
;  
; Parametry funkce:  
;   pA: ukazatel na pole A  
;   N: počet prvků pole A  
;   x: hodnota, kterou hledáme  
;  
; Návratová hodnota:  
;   EAX = 0, pokud je pA neplatný ukazatel (pA == 0) nebo pokud N <= 0 nebo pokud hodnota x nebyla v poli nalezena  
;   EAX = ukazatel na hledanou hodnotu (prvky pole jsou indexovány od 0)  
;  
; Důležité:  
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.  
;
```

```
;--- Úkol 2 ---  
;  
;  
; Naprogramujte funkci: int task22(const unsigned short *pA, int N, unsigned short x), která v poli 16bitových čísel bez znaménka pA  
; spočítá výskyt všech prvků pole stejných jako je hodnota x. Délka pole je dána parametrem N.  
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.  
;  
;  
; Parametry funkce:  
;   pA: ukazatel na pole A (pole obsahuje 16bitové hodnoty bez znaménka)  
;   N: počet prvků pole A (32bitové číslo se znaménkem)  
;   x: porovnávaná hodnota  
;  
;  
; Návratová hodnota:  
;   EAX = -1, když je pA neplatný ukazatel (pA == 0) nebo N < 0  
;   EAX = počet prvků pole stejných jako x  
;  
;  
; Důležité:  
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.  
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned int *pA, int N, unsigned int x) to search an array pA of N 32bit unsigned
; values for the first occurrence of the value x. The function returns position of the value in the array.
; The parameters are passed, the stack is cleaned and the result is returned according to the STDCALL calling convention.
;
; Function parameters:
;   pA: pointer to the array A to search in
;   N: length of the array A
;   x: value to be searched for
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N <= 0 or the value x has not been found in the array
;   EAX = position of the value x in the array (the array elements are indexed from 0)
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
%include "rw32-2020.inc"
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const unsigned int *pA, int N, unsigned int x), která v poli 32bitových čísel bez znaménka pA
; spočítá výskyt všech prvků pole větších než je hodnota x. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence PASCAL.
;
; Parametry funkce:
;   pA: ukazatel na pole A (pole obsahuje 32bitové hodnoty bez znaménka)
;   N: počet prvků pole A (32bitové číslo se znaménkem)
;   x: porovnávaná hodnota
;
; Návratová hodnota:
;   EAX = -1, když je pA neplatný ukazatel (pA == 0) nebo N < 0
;   EAX = počet prvků pole větších než x
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```



```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const unsigned char *pA, int N, unsigned char x), která v poli 8bitových čísel bez znaménka pA
; spočítá výskyt všech prvků pole menších než je hodnota x. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence STDCALL.
;
; Parametry funkce:
;   pA: ukazatel na pole A (pole obsahuje 8bitové hodnoty bez znaménka)
;   N: počet prvků pole A (32bitové číslo se znaménkem)
;   x: porovnávaná hodnota
;
; Návratová hodnota:
;   EAX = -1, když je pA neplatný ukazatel (pA == 0) nebo N < 0
;   EAX = počet prvků pole menších než x
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned short *pA, int N, unsigned short x) to count all occurrences of the values equal to x
; in an array pA of N 16bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array equal to x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const char *pA, int N, char x), která v poli 8bitových čísel se znaménkem pA
; spočítá výskyt všech prvků pole větších než je hodnota x. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.
;
; Parametry funkce:
;   pA: ukazatel na pole A (pole obsahuje 8bitové hodnoty se znaménkem)
;   N: počet prvků pole A (32bitové číslo se znaménkem)
;   x: porovnávaná hodnota
;
; Návratová hodnota:
;   EAX = -1, když je pA neplatný ukazatel (pA == 0) nebo N < 0
;   EAX = počet prvků pole větších než x
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const char *pA, int N, char x), která v poli 8bitových čísel se znaménkem pA
; o délce N nalezne poslední výskyt hodnoty x a vrátí index hledané hodnoty.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence STDCALL.
;
; Parametry funkce:
;   pA: ukazatel na pole A
;   N: počet prvků pole A
;   x: hodnota, kterou hledáme
;
; Návrátová hodnota:
;   EAX = -1, pokud je pA neplatný ukazatel (pA == 0) nebo pokud N <= 0 nebo pokud hodnota x nebyla v poli nalezena
;   EAX = index hledané hodnoty (prvky pole jsou indexovány od 0)
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned short *pA, int N, unsigned short x) to count all occurrences of the values equal to x
; in an array pA of N 16bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array equal to x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const unsigned int *pA, int N, unsigned int x), která v poli 32bitových čísel bez znaménka pA
; o délce N nalezne poslední výskyt hodnoty x a vrátí index hledané hodnoty.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.
;
; Parametry funkce:
;   pA: ukazatel na pole A
;   N: počet prvků pole A
;   x: hodnota, kterou hledáme
;
; Návratová hodnota:
;   EAX = -1, pokud je pA neplatný ukazatel (pA == 0) nebo pokud N <= 0 nebo pokud hodnota x nebyla v poli nalezena
;   EAX = index hledané hodnoty (prvky pole jsou indexovány od 0)
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const short *pA, int N, short x), která v poli 16bitových čísel se znaménkem pA
; o délce N nalezne poslední výskyt hodnoty x a vrátí index hledané hodnoty.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence PASCAL.
;
; Parametry funkce:
;   pA: ukazatel na pole A
;   N: počet prvků pole A
;   x: hodnota, kterou hledáme
;
; Návratová hodnota:
;   EAX = -1, pokud je pA neplatný ukazatel (pA == 0) nebo pokud N <= 0 nebo pokud hodnota x nebyla v poli nalezena
;   EAX = index hledané hodnoty (prvky pole jsou indexovány od 0)
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: unsigned short task22(const unsigned short *pA, int N), která v poli 16bitových čísel bez znaménka pA
; nalezne maximum a vrátí jeho hodnotu v registru AX. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence STDCALL.
;
; Parametry funkce:
;   pA: ukazatel na začátek pole 16bitových prvků bez znaménka
;   N: počet prvků pole pA (32bitové číslo se znaménkem)
;
; Návrátová hodnota:
;   EAX = 0x80000000, když je pA neplatný ukazatel (pA == 0) nebo N <= 0
;   AX = hodnota 16bitového maxima bez znaménka.
;
; Důležité:
;   Funkce musí zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```



```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: short task22(const short *pA, int N), která v poli 16bitových čísel se znaménkem pA
; nalezne maximum a vrátí jeho hodnotu v registru AX. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence STDCALL.
;
; Parametry funkce:
;   pA: ukazatel na začátek pole 16bitových prvků se znaménkem
;   N: počet prvků pole pA (32bitové číslo se znaménkem)
;
; Návratová hodnota:
;   EAX = 0x80000000, když je pA neplatný ukazatel (pA == 0) nebo N <= 0
;   AX = hodnota 16bitového maxima se znaménkem.
;
; Důležité:
;   Funkce musí zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const unsigned char *pA, int N, unsigned char x) to count all occurrences of the values lower than x
; in an array pA of N 8bit unsigned values.
; The parameters are passed, the stack is cleaned and the result is returned according to the STDCALL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;   x: comparison value
;
; Return values:
;   EAX = -1 if the pointer pA is invalid (pA == 0) or N < 0
;   EAX = count of the elements of the array lower than x
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: int task22(const char *pA, int N, char x), která v poli 8bitových čísel se znaménkem pA
; spočítá výskyt všech prvků pole větších než je hodnota x. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.
;
; Parametry funkce:
;   pA: ukazatel na pole A (pole obsahuje 8bitové hodnoty se znaménkem)
;   N: počet prvků pole A (32bitové číslo se znaménkem)
;   x: porovnávaná hodnota
;
; Návratová hodnota:
;   EAX = -1, když je pA neplatný ukazatel (pA == 0) nebo N < 0
;   EAX = počet prvků pole větších než x
;
; Důležité:
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Úkol 2 ---
;
; Naprogramujte funkci: unsigned int task22(const unsigned int *pA, int N), která v poli 32bitových čísel bez znaménka pA
; nalezne minimum a vrátí jeho hodnotu v registru EAX. Délka pole je dána parametrem N.
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.
;
; Parametry funkce:
;   pA: ukazatel na začátek pole 32bitových prvků bez znaménka
;   N: počet prvků pole pA (32bitové číslo se znaménkem)
;
; Návrátová hodnota:
;   EAX = 0x80000000, když je pA neplatný ukazatel (pA == 0) nebo N <= 0
;   EAX = hodnota 32bitového minima bez znaménka.
;
; Důležité:
;   Funkce musí zachovat obsah všech registrů, kromě registru EAX a příznakového registru.
;
```

```
;
;--- Task 2 ---
;
; Create a function: void* task22(const char *pA, int N, char x) to search an array pA of N 8bit signed
; values for the first occurrence of the value x. The function returns pointer to the value in the array.
; The parameters are passed, the stack is cleaned and the result is returned according to the CDECL calling convention.
;
; Function parameters:
;   pA: pointer to the array A to search in
;   N: length of the array A
;   x: value to be searched for
;
; Return values:
;   EAX = 0 if the pointer pA is invalid (pA == 0) or N <= 0 or the value x has not been found in the array
;   EAX = pointer to the value x in the array (the array elements are indexed from 0)
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

```
;--- Úkol 2 ---  
;  
; Naprogramujte funkci: int task22(const short *pA, int N, short x), která v poli 16bitových čísel se znaménkem pA  
; o délce N nalezne poslední výskyt hodnoty x a vrátí ukazatel na hledanou hodnotu.  
; Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence CDECL.  
;  
; Parametry funkce:  
;   pA: ukazatel na pole A  
;   N: počet prvků pole A  
;   x: hodnota, kterou hledáme  
;  
; Návratová hodnota:  
;   EAX = 0, pokud je pA neplatný ukazatel (pA == 0) nebo pokud N <= 0 nebo pokud hodnota x nebyla v poli nalezena  
;   EAX = ukazatel na hledanou hodnotu (prvky pole jsou indexovány od 0)  
;  
; Důležité:  
;   - funkce MUSÍ zachovat obsah všech registrů, kromě registru EAX a příznakového registru.  
;
```

```
;
;--- Task 2 ---
;
; Create a function: int task22(const int *pA, int N) to find maximum in an array pA of N 32bit signed values.
; The parameters are passed, the stack is cleaned and the result is returned according to the PASCAL calling convention.
;
; Function parameters:
;   pA: pointer to the array A
;   N: length of the array A
;
; Return values:
;   EAX = 0x80000000 if the pointer pA is invalid (pA == 0) or N <= 0
;   EAX = value of the 32bit signed maximum
;
; Important:
;   - the function MUST preserve content of all the registers except for the EAX and flags registers.
;
```

--- Úkol 2 ---

Naprogramujte funkci: `int task22(const unsigned char *pA, int N, unsigned char x)`, která v poli 8bitových čísel bez znaménka `pA` spočítá výskyt všech prvků pole stejných jako je hodnota `x`. Délka pole je dána parametrem `N`. Funkce dostává parametry, uklízí zásobník a vrací výsledek podle konvence PASCAL.

Parametry funkce:

- `pA`: ukazatel na pole `A` (pole obsahuje 8bitové hodnoty bez znaménka)
- `N`: počet prvků pole `A` (32bitové číslo se znaménkem)
- `x`: porovnávaná hodnota

Návratová hodnota:

- `EAX = -1`, když je `pA` neplatný ukazatel (`pA == 0`) nebo `N < 0`
- `EAX = počet prvků pole stejných jako x`

Důležité:

- funkce MUSÍ zachovat obsah všech registrů, kromě registru `EAX` a příznakového registru.