[Ex5]給助教的

> 408261292 丁柏瑋

要是這邊的註解沒對齊，那是字形問題。

C++

```
template <typename T>
void insertSort(T nums[],int arraySize)
{
    for(int i=1; i<arraySize; i++)
        for(int j=i; j>0&&nums[j]<nums[j-1]; j--)
            swap(nums[j],nums[j-1]);
}
```

Assembly

libary.inc

```
INCLUDE Irvine32.inc

;C library
includelib ucrt.lib
includelib legacy_stdio_definitions.lib


scanf PROTO C,format:PTR BYTE, args:VARARG

insertionSort PROTO,
  ptrArray:PTR SDWORD,arraySize:DWORD,ptrBlank:PTR BYTE

arrayOutput PROTO,
  ptrArray:PTR SDWORD,arraySize:DWORD,ptrBlank:PTR BYTE

outputCenter PROTO,
  ptrArray:PTR SDWORD,arraySize:DWORD,ptrBlank:PTR BYTE

ExitProcess PROTO,dwExitCode:DWORD
```

Main.asm

```
;-----------------------------------------------
;Exercise-5: Implementation of Sorting Algorithms(Insertion sort)
;Author : 408261292
;
; Implement a sorting algorithm to sort a set of integers in ascending order.
;-----------------------------------------------

INCLUDE libary.inc

.data
  array SDWORD 200 DUP (0)
  arraySize DWORD 0
  intFormat BYTE "%d",0
  blank BYTE " ",0
.code
main proc

proc_begin:

  ;input arraySize and array data.

  invoke scanf,ADDR intFormat,ADDR arraySize
  mov ecx,arraySize
  mov eax,arraySize
  cmp eax,0
  je proc_end
  mov esi,OFFSET array                    ;set array offset

data_input:
  pushad
  invoke scanf,ADDR intFormat,ADDR [esi]
  popad
  add esi,TYPE Array                      ;next SDWORD space
  loop data_input

  ;input data end

  ;sort function by invoke
  INVOKE insertionSort,ADDR array,arraySize,ADDR blank

  ;Output array by invoke
  call crlf
  INVOKE arrayOutput,ADDR array,arraySize,ADDR blank
  INVOKE outputCenter,ADDR array,arraySize,ADDR blank

  jmp proc_begin
proc_end:
  invoke ExitProcess,0
main endp
end main
```

insertionSort.asm

```
;-------------------------------------------------
;Sort by insertion sort
;
;C sample:
;for(int i=1; i<arraySize; i++)
;    for(int j=i; j>0&&nums[j]<nums[j-1]; j--)
;        swap(nums[j],nums[j-1]);
;-------------------------------------------------

INCLUDE libary.inc

.code
insertionSort PROC,
  ptrArray:PTR SDWORD,arraySize:DWORD,ptrBlank:PTR BYTE

  mov esi,ptrArray
  mov eax,1                         ;eax=i=1

for1_begin:
  ;for1 title
  cmp eax,arraySize                     ;i<arraySize
  jnb for1_end;
  push eax                          ;i in stack

  ;for1 body
  for2_begin:
    ;for2 title
    ;j>0 && nums[j]<nums[j-1]
    cmp eax,0
    jna for2_end

    ;nums[j]<nums[j-1]
    mov ebx,[esi+4*eax]
    dec eax
    cmp ebx,[esi+4*eax]
    jnl for2_end
    inc eax
    push eax                        ;j in stack

    ;swap(nums[j],nums[j-1]);
    mov ebx,[esi+4*eax]
    dec eax
    xchg ebx,[esi+4*eax]
    inc eax
    mov [esi+4*eax],ebx

    ;for2 tail
    pop eax
    dec eax
    jmp for2_begin
  for2_end:
  pushad
  INVOKE arrayOutput,ptrArray,arraySize,ptrBlank
  popad
  ;for1 tail
  pop eax
```

```
  inc eax
  jmp for1_begin
for1_end:

  ret
  insertionSort ENDP
END
```

arrayOutput.asm

```
INCLUDE libary.inc

.code
arrayOutput PROC,
  ptrArray:PTR SDWORD,arraySize:DWORD,ptrBlank:PTR BYTE

  mov esi,ptrarray
  mov ecx,arraySize
  mov edx,ptrBlank
next_out:
  mov eax,[esi]
  cmp eax,0
  jl negative

postive:                ;+
  call writeDec
  jmp numberOut_end

negative:               ;-
  call writeInt

numberOut_end:

  call writeString
  add esi,4
  loop next_out
  mov esi,ptrarray
  call crlf

  ret
  arrayOutput ENDP


outputCenter PROC,
  ptrArray:PTR SDWORD,arraySize:DWORD,ptrBlank:PTR BYTE

  mov esi,ptrArray
  mov eax,arraySize
  mov ebx,2
  mov edx,0
  div ebx
  mov ecx,eax

toCenter:
```

```
  ;move ptr to center
  add esi,4
  loop toCenter
  mov eax,[esi]

  cmp eax,0
  jl negative2

postive2:               ;+
  call writeDec
  jmp numberOut_end2

negative2:                ;-
  call writeInt

numberOut_end2:

  ;if arraySize==even,output [esi-4]
  cmp edx,0
  jne center_end

  mov edx,ptrBlank
  call writeString

  sub esi,4
  mov eax,[esi]

  cmp eax,0
  jl negative3

postive3:               ;+
  call writeDec
  jmp numberOut_end3

negative3:                ;-
  call writeInt

numberOut_end3:

center_end:

    call crlf

    ret
  outputCenter ENDP
END
```