ITCS222

Computer Organization and Architecture

Programming Project 1

Waris Damkham 6388014

Section 1

Assoc. Prof. Dr. Chomtip Pornpanomchai

Ascending sort three integers: C language

Code:

```c
#include <stdio.h>
void swap(int *num1,int *num2)
{
  int tmp = *num1;
  *num1 = *num2;
  *num2 = tmp;
}
int main(void)
{
  int num1,num2,num3;
  printf("input the first number:");
  scanf("%d",&num1);
  printf("input the second number:");
  scanf("%d",&num2);
  printf("input the third number:");
  scanf("%d",&num3);
  if(num1>num3)
  {
    swap(&num1,&num3);
  }
  if(num1>num2)
  {
    swap(&num1,&num2);
  }
  if(num2>num3)
  {
    swap(&num2,&num3);
  }
  printf("After the Ascending Sorting : %d %d %d",num1,num2,num3);
  return 0;
}
```

The Output:

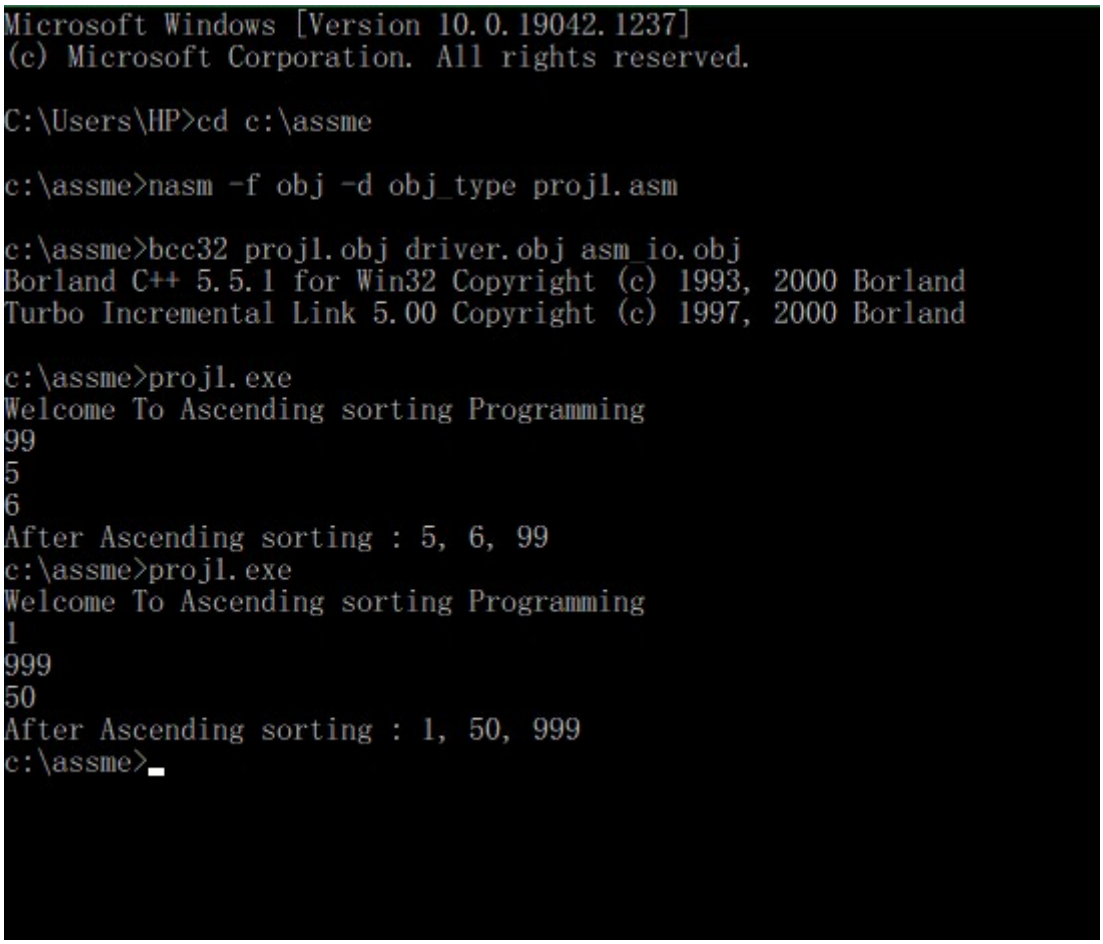
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\VS\cc++> cd "d:\VS\cc++\" ; if ($?) { gcc Projectarc.c -o Projectarc } ; if ($?) { .\Projectarc }
input the first number:99
input the second number:120
input the third number:19
After the Ascending Sorting : 19 99 120
PS D:\VS\cc++>
```

Ascending sort three integers: Assembly language

Code:

```
;Waris Damkham 6388014
%include "asm_io.inc"

segment _DATA public align=4 class=DATA use32
M0          db       "Welcome To Ascending sorting Programming", 0ah, 0dh, 0
M1          db       "After Ascending sorting : ", 0
M2          db       ", ",0

segment _BSS public align=4 class=BSS use32
F           resd    1
S           resd    1
T           resd    1
TMP         resd    1

group DGROUP _BSS _DATA

segment _TEXT public align=1 class=CODE use32
        global  _asm_main
_asm_main:
        enter   0,0
        pusha
        mov     eax,M0
        call    print_string

        call    read_int
        mov     [F], eax

        call    read_int
        mov     [S], eax

        call    read_int
        mov     [T], eax

        mov     eax, [F]
        cmp     eax, [S]
        jg      SWAP_XY
        mov     eax, [S]
        cmp     eax, [T]
        jg      SWAP_YZ
        jmp     PROCESS

SWAP_XY:
    mov eax, [F]
    mov [TMP], eax
    mov eax, [S]
    mov [F], eax
    mov eax, [TMP]
    mov [S], eax
    jmp PROCESS

PROCESS:
    mov     eax, [S]
    cmp     eax, [T]
    jg      SWAP_YZ
    mov     eax,  M1
    call    print_string
    mov     eax, [F]
    call    print_int
    mov     eax,M2
    call    print_string
    mov     eax, [S]
    call    print_int
    mov     eax,M2
    call    print_string
    mov     eax, [T]
    call    print_int
    jmp     END

SWAP_YZ:
    mov     eax, [S]
    mov     [TMP], eax
    mov     eax, [T]
    mov     [S], eax
    mov     eax, [TMP]
    mov     [T], eax
    mov     eax, [F]
    cmp     eax, [S]
    jg      SWAP_XY
    mov     eax,  M1
    call    print_string
    mov     eax, [F]
    call    print_int
    mov     eax,M2
    call    print_string
    mov     eax, [S]
    call    print_int
    mov     eax,M2
    call    print_string
    mov     eax, [T]
    call    print_int
    jmp     END

END:
    popa
    mov eax, 0
    leave
    ret
```

The Output:


Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.
C:\Users\HP>cd c:\assme
c:\assme>nasm -f obj -d obj_type proj1.asm
c:\assme>bcc32 proj1.obj driver.obj asm_io.obj
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland
c:\assme>proj1.exe
Welcome To Ascending sorting Programming
99
5
6
After Ascending sorting : 5, 6, 99
c:\assme>proj1.exe
Welcome To Ascending sorting Programming
1
999
50
After Ascending sorting : 1, 50, 999
c:\assme>