

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:

```
1  #include <stdio.h>
2  void swap(int *num1,int *num2)
3  {
4      int tmp = *num1;
5      *num1 = *num2;
6      *num2 = tmp;
7  }
8  int main(void)
9  {
10     int num1,num2,num3;
11     printf("input the first number:");
12     scanf("%d",&num1);
13     printf("input the second number:");
14     scanf("%d",&num2);
15     printf("input the third number:");
16     scanf("%d",&num3);
17     if(num1>num3)
18     {
19         swap(&num1,&num3);
20     }
21     if(num1>num2)
22     {
23         swap(&num1,&num2);
24     }
25     if(num2>num3)
26     {
27         swap(&num2,&num3);
28     }
29     printf("After the Ascending Sorting : %d %d %d",num1,num2,num3);
30     return 0;
31 }
```

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:

```
1 ;Waris Damkham 6388014
2 %include "asm_io.inc"
3
4 segment _DATA public align=4 class=DATA use32
5 M0 db "Welcome To Ascending sorting Programming", 0ah, 0dh, 0
6 M1 db "After Ascending sorting : ", 0
7 M2 db ", ", 0
8
9 segment _BSS public align=4 class=BSS use32
10 F resd 1
11 S resd 1
12 T resd 1
13 TMP resd 1
14
15 group DGROUP _BSS _DATA
16
17 segment _TEXT public align=1 class=CODE use32
18 global _asm_main
19
20 _asm_main:
21     enter 0,0
22     pusha
23     mov eax,M0
24     call print_string
25
26     call read_int
27     mov [F], eax
28
29     call read_int
30     mov [S], eax
31
32     call read_int
33     mov [T], eax
34
35     mov eax, [F]
36     cmp eax, [S]
37     jg SWAP_XY
38     mov eax, [S]
39     cmp eax, [T]
40     jg SWAP_YZ
41     jmp PROCESS
42
43 SWAP_XY:
44     mov eax, [F]
45     mov [TMP], eax
46     mov eax, [S]
47     mov [F], eax
48     mov eax, [TMP]
49     mov [S], eax
50     jmp PROCESS
51
52 PROCESS:
53     mov eax, [S]
54     cmp eax, [T]
55     jg SWAP_YZ
56     mov eax, M1
57     call print_string
58     mov eax, [F]
59     call print_int
60     mov eax, M2
61     call print_string
62     mov eax, [S]
63     call print_int
64     mov eax, M2
65     call print_string
66     mov eax, [T]
67     call print_int
68     jmp END
69
70 SWAP_YZ:
71     mov eax, [S]
72     mov [TMP], eax
73     mov eax, [T]
74     mov [S], eax
75     mov eax, [TMP]
76     mov [T], eax
77     mov eax, [F]
78     cmp eax, [S]
79     jg SWAP_XY
80     mov eax, M1
81     call print_string
82     mov eax, [F]
83     call print_int
84     mov eax, M2
85     call print_string
86     mov eax, [S]
87     call print_int
88     mov eax, M2
89     call print_string
90     mov eax, [T]
91     call print_int
92     jmp END
93
94 END:
95     popa
96     mov eax, 0
97     leave
98     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>_
```