

Qno 1

```
1
2 INCLUDE Irvine32.inc
3
4 .data
5 sizee dword 10
6 array SDWORD SIZEOF sizee DUP(30,-40,20,65,80,45)
7 j DWORD ?
8 k DWORD ?
9
10
11 .code
12
13     main PROC
14         MOV j, 20
15         MOV k, 50
16         MOV ESI, OFFSET array
17         MOV ECX, sizee
18         CALL ArraySum
19         CALL WriteInt
20         CALL crlf
21
22         MOV j, 35
23         MOV k, 90
24         MOV ESI, OFFSET array
25         MOV ECX, sizee
26         CALL ArraySum
27         CALL WriteInt
28         CALL crlf
29     exit
30
31     main ENDP
32
33 ArraySum PROC USES ecx esi
34     MOV eax, 0
35
36     L1:
37         MOV ebx, [esi]
38         cmp ebx, j
39         jge loop11
40         jmp trvs
41
42     loop11:
43         cmp ebx, k
44         jle loop22
45         jmp trvs
46
47     loop22:
48         add eax, ebx
49
50     trvs:
51         add esi, 4
52
53     loop L1
54     ret
55 ArraySum ENDP
56
57 END main
```

Qno 2 part a

```
1 INCLUDE Irvine32.inc
2 .data
3     msg1 byte "Enter a single-digit number: ", 0
4     msg2 byte "Sorted Array is: ", 0
5     array byte 10 Dup(?)
6 .code
7
8     main PROC
9         mov ecx, LENGTHOF array
10        mov al, 0
11        mov esi, 0
12        L1:
13
14            mov edx, OFFSET msg1
15            call WriteString
16            call ReadDec
17            mov DWORD PTR array[esi*TYPE array], eax
18            inc esi
19            loop L1
20        mov ecx, OFFSET array
21
22        push ecx
23
24        mov esi, OFFSET array
25
26        mov ecx, LENGTHOF array
27
28        call FindMinimum
29
30
31
32        mov edi, OFFSET array
33
34        mov ecx, LENGTHOF array
35
36        call SelectionSort
37
38        mov ecx, OFFSET array
39
40        push ecx
41
42        mov edx, OFFSET msg2
43
44        call WriteString
45
46        call CRLF
47
48        call DisplayArray
49        exit
50 main ENDP
51
52
53 DisplayArray PROC
54
55     pushad
```

Qno 2 part b

```
56
57     mov eax, 0
58
59     mov ecx, LENGTHOF array
60
61     mov esi, 0
62
63     L2:
64
65         mov al, array[esi]
66
67         call WriteDec
68
69         mov al, TAB
70
71         call WriteChar
72
73         inc esi
74
75         call Crlf
76
77
78
79     loop L2
80
81     popad
82
83     call Crlf
84
85     ret
86
87
88
89 DisplayArray ENDP
90
91 SelectionSort PROC
92
93     dec ecx
94
95     mov ebx, edi
96
97     mov edx, ecx
98     OuterLoop:
99
100         mov edi, ebx
101
102         mov esi, edi
103
104         inc esi
105
106         push ecx
107
108         mov ecx, edx
109
110
111
```

Qno 2 part c

```
110
111         mov al, [esi]
112
113         cmp al, [edi]
114
115         pushf
116
117         inc esi
118
119         inc edi
120
121         popf
122
123         jae doNotSwap
124
125         call Swap
126         doNotSwap:
127
128         loop InnerLoop
129         pop ecx
130
131         loop OuterLoop
132     ret
133 SelectionSort ENDP
134
135 FindMinimum PROC
136
137     mov edi, esi
138     minimumIndex:
139
140         mov al, [esi]
141
142         cmp al, [edi]
143
144         jae skip
145
146         mov edi, esi
147     skip:
148
149         inc esi
150         loop minimumIndex
151     ret
152 FindMinimum ENDP
153 Swap PROC
154
155     mov al, [esi-1]
156
157     mov ah, [edi-1]
158
159     mov [esi-1], ah
160
161     mov [edi-1], al
162
163     ret
164 Swap ENDP
165 END main
```

Qno 3 Part a

```
1 INCLUDE irvine32.inc
2
3
4 .data
5
6 msg1 byte "Enter a single-digit number: ", 0
7 msg2 byte "Sorted Array is: ", 0
8 array DWORD 10 Dup(?)
9
10
11 .code
12     main PROC
13
14
15     mov ecx, LENGTHOF array
16     mov al, 0
17     mov esi, 0
18     L1:
19         mov edx, OFFSET msg1
20         call WriteString
21         call ReadDec
22         mov array[esi*TYPE array], eax
23         inc esi
24
25         loop L1
26
27
28     push LENGTHOF array
29     push OFFSET array
30     call BUBBLE
31
32
33     exit
34     main ENDP
35
36     BUBBLE PROC
37         push ebp
38         mov ebp, esp
39
40         mov esi, [ebp+8]
41         mov edi, esi
42         mov ebx, [ebp+12]
43         mov ecx, ebx
44         mov edi, 0
45
46         L1:
47             mov edx, ecx
48             mov ecx, 9
49             L2:
50                 mov eax, 0
51                 mov ebx, 0
52                 mov eax, array[edi * TYPE array]
53                 inc edi
54                 mov ebx, array[edi * TYPE array]
55                 dec edi
```


Qno 3 Part b

```
56
57             cmp eax, ebx
58             JG call_swap
59             back:
60             inc edi
61             loop L2
62
63             mov edi, 0
64             inc esi
65             mov ecx, edx
66
67             loop L1
68
69             call Print_Array
70
71             ret
72
73             call_swap:
74                 call swap
75                 jmp back
76
77             mov esp, ebp
78             pop ebp
79
80             BUBBLE ENDP
81
82             swap PROC
83                 xchg eax, ebx
84                 mov array[edi * TYPE array], eax
85                 inc edi
86                 mov array[edi * TYPE array], ebx
87                 dec edi
88                 mov eax, 0
89                 mov ebx, 0
90
91                 ret
92             swap ENDP
93
94             Print_Array PROC
95                 mov esi, 0
96                 mov ecx, 10
97                 mov edx, OFFSET msg2
98                 call WriteString
99                 call CRLF
100                mov edx, 0
101
102                L3:
103                    mov eax, array[edx * TYPE array]
104                    call writedec
105                    call CRLF
106                    inc edx
107                loop L3
108                ret
109             Print_Array endp
110             END main
```

Qno 4

```
1 INCLUDE irvine32.inc
2
3 .data
4
5 msg1 byte "Enter a Number: ", 0
6 msg2 byte "Factorial of the Number is: ", 0
7
8 .code
9     main PROC
10         MOV edx, OFFSET msg1
11         CALL WriteString
12         MOV eax, 0
13         CALL Readdec
14         MOV ecx, eax
15         MOV eax, 0
16         PUSH ecx
17         CALL FACTORIAL
18
19     exit
20     main ENDP
21
22
23 FACTORIAL PROC
24     PUSH ebp
25     MOV ebp, esp
26
27     MOV ecx, [ebp+8]
28     MOV eax, ecx
29
30     L1:
31         MOV ebx, ecx
32         SUB ebx, 1
33         CMP ebx, 0
34         JG fac
35         JMP move
36         fac:
37             MUL ebx
38
39     move:
40         LOOP L1
41
42     CALL CRLF
43     MOV edx, OFFSET msg2
44     CALL WriteString
45     CALL WriteDec
46     CALL CRLF
47
48
49     MOV esp, ebp
50     POP ebp
51
52     RET
53 FACTORIAL ENDP
54
55 END main
```

Qno 5

```
1
2 INCLUDE Irvine32.inc
3
4 .data
5
6 char    BYTE    "Enter any character: ",0
7 ascii   BYTE    "ASCII value Of the Character: ",0
8 count   BYTE    "1's Count: ",0
9
10 .code
11
12 main PROC
13
14     MOV eax,0
15     LEA edx,char
16     CALL writestring
17     CALL readchar
18     CALL writechar
19     CALL crlf
20
21
22     MOV bl,00000000b
23     AND ah,bl
24     LEA edx,ascii
25     CALL writestring
26     CALL writebin
27     CALL crlf
28
29     MOV ecx,8
30     MOV ebx,0
31
32     L1:
33         SHR al,1
34         JNC backL1
35         ADD ebx,1
36     backL1:
37         LOOP L1
38
39     LEA edx,count
40     CALL writestring
41     MOV eax,ebx
42     CALL writedec
43
44
45
46 exit
47 main ENDP
48 END main
```


Qno 6

```
1 INCLUDE Irvine32.inc
2
3 .data
4
5
6     arr1 SDWORD 1,2,3,4,5,6,7,8
7     arr2 SDWORD 5,6,7,8,9,10,11,12
8     counter DWORD 0
9
10    .code
11
12    main PROC
13        INVOKE CountNearMatches, ADDR arr1, ADDR arr1, LENGTHOF arr1
14        call WriteInt
15        call Crlf
16
17        exit
18    main ENDP
19
20
21 CountNearMatches PROTO, ADDR arr1, ADDR arr1, LENGTHOF arr1
22     mov esi,ptrarr1
23     mov edi,ptrarr2
24     mov ecx,sizearr
25
26 L1:
27     mov ebx,0
28     mov ebx,[esi]
29     mov edx,0
30     mov edx,[edi]
31     cmp ebx,edx
32     jne equal
33     inc counter
34     jmp here
35
36 equal:
37     jmp here
38
39 here:
40     add esi, SIZEOF SDWORD
41     add edi, SIZEOF SDWORD
42
43     loop L1
44 ; increment count
45     mov eax,0
46     mov eax,counter
47     mov counter,0
48
49     ret
50 CountNearMatches ENDP
51
52 END main
```

Qno 7

```
1 INCLUDE Irvine32.inc
2 .data
3
4 diff DWORD 3 DUP(?)
5
6 Number1 QWORD 1010101010b
7 Number2 QWORD 1111000001b
8
9
10 .code
11
12 main PROC
13
14 MOV esi, offset Number1
15 MOV edi, offset Number2
16
17 MOV ebx, OFFSET diff
18 MOV ecx, 2
19
20 CALL Extended_Sub
21
22 CALL WriteBin
23 EXIT
24 main ENDP
25
26 Extended_Sub proc
27
28 PUSHAD
29 cld
30
31     L1:
32
33     MOV eax,[esi]
34     SBB eax,[edi]
35     PUSHFD
36     MOV [ebx],eax
37     ADD esi, TYPE num1
38     ADD edi, TYPE num2
39     SUB ebx, 4
40     POPFD
41     LOOP L1
42
43 SBB word ptr [ebx],0
44 POPAD
45
46 MOV eax,[ebx]
47 RET
48 |
49 Extended_sub ENDP
50
51
52 END MAIN
53
```

Qno 8

```
1 INCLUDE Irvine32.inc
2
3
4 .data
5
6 Number1 QWORD 10101010b
7 Number2 QWORD 10010000b
8 Result Dword 3 dup(?)
9
10 .code
11
12 main PROC
13
14     MOV edi, offset Number2
15     MOV esi, offset Number1
16     MOV ebx, Offset Result
17     MOV ecx, 2
18     CALL Extended_ADD
19     CALL writebin
20
21 EXIT
22 main ENDP
23
24 Extended_ADD proc
25     PUSHAD
26     cld
27     L1:
28         MOV eax,[esi]
29         ADC eax,[edi]
30         pushfd
31         MOV [ebx],eax
32         add esi, TYPE Number1
33         add edi, TYPE Number2
34         add ebx, 4
35         popfd
36         LOOP L1
37
38     ADC WORD PTR[ebx],0
39     POPAD
40     MOV eax,[ebx]
41     RET
42 Extended_ADD endp
43
44 END main
45
```