

# Practice God Level Questions 2

## Exercise 1: Draw Text Colors

```
598 include Irvine32.inc
599
600 .data
601     colors BYTE 2, 4, 6, 14    ; Colors: Green, Red, Yellow, White
602     message BYTE "Hello, Colors!",0
603
604 .code
605 main PROC
606     mov ecx, 4    ; Number of colors
607     mov esi, 0    ; Index for colors array
608
609 loop_colors:
610     mov eax, colors[esi]
611     call SetTextColor
612     mov edx, OFFSET message
613     call WriteString
614     call Crlf
615
616     inc esi
617     loop loop_colors
618
619     call WaitMsg    ; Wait for a key press
620     call Clrscr     ; Clear the screen
621     call ExitProcess
622 main ENDP
623 END main
```

## Exercise 2: Linking Array Items

```

.data
start DWORD 1
chars BYTE 'H', 'A', 'C', 'E', 'B', 'D', 'F', 'G'
links DWORD 0, 4, 5, 6, 2, 3, 7, 0
outputArray BYTE 8 DUP(?) ; To store the characters in order

.code
main PROC
    mov edi, OFFSET outputArray ; Destination for output characters
    mov esi, start ; Start index
    mov ecx, 8 ; Number of characters to locate
    traverse_links:
        mov al, chars[esi] ; Load character
        mov [edi], al ; Store it in outputArray
        inc edi ; Move to the next position in outputArray
        ; Get the next link index
        mov eax, esi
        mov ebx, 4 ; Size of DWORD (4 bytes)
        mul ebx ; Multiply esi by 4
        mov esi, links[eax] ; Get the next link index

        loop traverse_links ; Repeat for all characters

    ; Display the characters in outputArray
    mov edx, OFFSET outputArray
    call WriteString
    call Crlf

    call WaitMsg ; Wait for a key press
    call ExitProcess

main ENDP
END main

```

### Exercise 3: Simple Addition (1)

```

662 include Irvine32.inc
663 .data
664     prompt1 BYTE "Enter the first integer: ", 0
665     prompt2 BYTE "Enter the second integer: ", 0
666     resultMsg BYTE "The sum is: ", 0
667     buffer1 DWORD ?
668     buffer2 DWORD ?
669     result DWORD ?
670 .code
671     main PROC
672         call Clrscr
673         mov edx, OFFSET prompt1
674         call WriteString
675         call ReadInt
676         mov buffer1, eax
677
678         mov edx, OFFSET prompt2
679         call WriteString
680         call ReadInt
681         mov buffer2, eax
682
683         ; Add the integers
684         mov eax, buffer1
685         add eax, buffer2
686         mov result, eax
687
688         mov edx, OFFSET resultMsg
689         call WriteString
690         mov eax, result
691         call WriteInt
692
693         call WaitMsg ; Wait for a key press
694         call ExitProcess
695     main ENDP
696 END main

```

```

700 include Irvine32.inc
701 .data
702     prompt1 BYTE "Enter the first integer: ", 0
703     prompt2 BYTE "Enter the second integer: ", 0
704     resultMsg BYTE "The sum is: ", 0
705     buffer1 DWORD ?
706     buffer2 DWORD ?
707     result DWORD ?
708
709 .code
710     main PROC
711         mov ecx, 3 ; Repeat the process three times
712         loop_repeat:
713             call Clrscr
714
715             mov edx, OFFSET prompt1
716             call WriteString
717             call ReadInt
718             mov buffer1, eax
719
720             mov edx, OFFSET prompt2
721             call WriteString
722             call ReadInt
723             mov buffer2, eax
724
725             ; Add the integers
726             mov eax, buffer1
727             add eax, buffer2
728             mov result, eax

```

```

729
730     mov edx, OFFSET resultMsg
731     call WriteString
732     mov eax, result
733     call WriteInt
734
735     call WaitMsg ; Wait for a key press
736
737     dec ecx
738     jnz loop_repeat ; Repeat the process three times
739
740     call ExitProcess
741 main ENDP
742 END main

```

### ***Exercise 5: BetterRandomRange Procedure***

```

746 include Irvine32.inc
747
748 .data
749
750 .code
751 BetterRandomRange PROC
752     ; Input: EBX = Lower bound (M)
753     ;         EAX = Upper bound (N)
754     ; Output: EAX = Random number between M and N-1
755
756     sub eax, ebx ; Calculate the range (N - M)
757     add eax, 1   ; Include the upper bound itself
758     call RandomRange
759     add eax, ebx ; Offset the result by M (lower bound)
760     ret
761 BetterRandomRange ENDP
762
763 main PROC
764     mov ecx, 50 ; Repeat 50 times
765     loop_repeat:
766         call Clrscr
767
768         mov ebx, -300 ; Lower bound
769         mov eax, 100 ; Upper bound
770         call BetterRandomRange
771
772         ; Display the randomly generated value
773         mov edx, eax
774         call WriteInt
775         call Crlf
776
777         call WaitMsg ; Wait for a key press
778         dec ecx
779         jnz loop_repeat ; Repeat the process 50 times
780
781     call ExitProcess
782 main ENDP
783 END main

```

These solutions demonstrate the requested functionality for each exercise. Feel free to modify them, as you do your practice to be a

better programmer.