```
ASSEMBLLY CODE:
    .MODEL SMALL
    .STACK 100H
    .DATA
        MSGA DB 'First Choice: $'
        MSGB DB 'Second Choice: $'
        MSGC DB 'Third Choice: $'
        MSGD DB 'The Diffrent Choice Winer: $'
        P_S DB '* $'
        P_A DB 'A $'
        p_B DB 'B $'
        P_C DB 'C $'
        NEW_L DB 13,10,'$'
        COUNT DB ?
        A DW ?
        B DW ?
        C DW ?
        TEMP DW ?

    .CODE
    MAIN PROC
        MOV AX,@DATA
        MOV DS,AX

    RAND_IO:
        MOV COUNT,3
        LEA DX,MSGA                     ;Print 'First Choice:'
        MOV AH,9
        INT 21H

        GET_INPUTS:
            MOV DX,0
            MOV BX,0                    ;Clear bx
            MOV AH,1
            INT 21H
            CMP AL,0DH
            JE END_INPUTS               ;If enter

            CONVERT_TO_NUM:
                AND AX,000FH            ;Use full 16 bits of AX
                MOV TEMP,AX
                MOV AX,10
                MUL BX
                MOV BX, AX
                ADD BX,TEMP

            MOV AH,1                    ;Input new digit
            INT 21H
            CMP AL,0DH
            JNE CONVERT_TO_NUM

            END_INPUTS:
                CMP COUNT,3
                JE GET_A

                CMP COUNT,2
                JE GET_B

                CMP COUNT,1
                JE GET_C

                GET_A:
                    MOV A,BX
                    LEA DX,MSGB         ;Print 'Second choice: '
                    MOV AH,9
                    INT 21H
                    DEC COUNT
                    JMP GET_INPUTS


                GET_B:
                    MOV B,BX
                    LEA DX,MSGC                 ;Print 'Third choice: '
                    MOV AH,9
                    INT 21H
                    DEC COUNT
                    JMP GET_INPUTS

                GET_C:
                    MOV C,BX

        LEA DX,MSGD                     ;print The Diffrent Choice Winer:
        MOV AH,9
        INT 21H

        GET_RESULTS:
            MOV AX,A
            CMP AX,B                    ;Compaire A and B
            JE CMP_ABC                  ;if A+B print C

                MOV BX,B
                CMP BX,C                ;Compair B and C
                JE PRINT_A              ;if b = C print A

                    MOV AX,A
                    CMP AX,C            ;Compair A and C
                    JE PRINT_B          ;if A = C print B

                    JMP NEW_INPUT

            CMP_ABC:
                CMP AX,C                ;Compair B & C after checking A = B
                JE PRINT_STAR
                JMP PRINT_C

        PRINT_A:
            LEA DX,P_A
            MOV AH,9
            INT 21H
            JMP NEW_INPUT

        PRINT_B:
            LEA DX,P_B
            MOV AH,9
            INT 21H
            JMP NEW_INPUT

        PRINT_C:
            LEA DX,P_C
            MOV AH,9
            INT 21H
            JMP NEW_INPUT

        PRINT_STAR:
            LEA DX,P_S
            MOV AH,9
            INT 21H
            JMP NEW_INPUT

        NEW_INPUT:
            LEA DX,NEW_L
```

```
                MOV AH,9
                INT 21H
                JMP RAND_IO

        MAIN ENDP
            END MAIN
```