Test Cases and Marking Scheme

Question 1: Accumulative Array (20 marks)

[4 points every case]

CASE 1:

```
Please enter integers in array A[] one by one:
A[0]: 1
A[1]: 2
A[2]: 3
A[3]: 4
A[4]: 5
A[5]: 6
A[6]: 7
A[7]: 8
A[8]: 9
A[9]: 10
Here is the accumulative array of A:
1 3 6 10 15 21 28 36 45 55
-- program is finished running --
```

CASE 2:

```
Please enter integers in array A[] one by one:
A[0]: 0
A[1]: 0
A[2]: 0
A[3]: 0
A[4]: 0
A[5]: 0
A[6]: 0
A[7]: 0
A[8]: 0
A[9]: 0
Here is the accumulative array of A:
0 0 0 0 0 0 0 0 0 0
-- program is finished running --
```

CASE 3:

```
Please enter integers in array A[] one by one:
A[0]: 1
A[1]: -1
A[2]: 2
A[3]: -2
A[4]: 3
A[5]: -3
A[6]: 4
A[7]: -4
A[8]: 5
A[9]: -5
Here is the accumulative array of A:
1 0 2 0 3 0 4 0 5 0
-- program is finished running --
```

CASE 4:

```
Please enter integers in array A[] one by one:
A[0]: -1
A[1]: -2
A[2]: -3
A[3]: -4
A[4]: -5
A[5]: -6
A[6]: -7
A[7]: -8
A[8]: -9
A[9]: -10
Here is the accumulative array of A:
-1 -3 -6 -10 -15 -21 -28 -36 -45 -55
-- program is finished running --
```

CASE 5:

Change this lines in student's code

```
# array A[] has 10 elements, each element is of the size of one word, 32
bits.
A: .word 0:10
# array Accum[], 10 elements.
Accum: .word 0:10
# size: the number of elements in array A[]
size: .word 10
```

to

```
A: .word 0:15
Accum: .word 0:15
size: .word 15
```

```
Please enter integers in array A[] one by one:
A[0]: 1
A[1]: -2
A[2]: 3
A[3]: -4
A[4]: 5
A[5]: -6
A[6]: 7
A[7]: -8
A[8]: 9
A[9]: -10
A[10]: 11
A[11]: -12
A[12]: 13
A[13]: -14
A[14]: 15
Here is the accumulative array of A:
1 -1 2 -2 3 -3 4 -4 5 -5 6 -6 7 -7 8
-- program is finished running --
```

Question 2: Skew-symmetric Matrix (20 marks)

[4 points every case]

CASE 1:

```
Please enter the size of your matrix:
3
Please enter integers in array A[] one by one:
A[0][0]: 0
A[0][1]: 2
A[0][2]: -45
A[1][0]: -2
A[1][1]: 0
A[1][2]: 4
A[2][0]: 45
A[2][1]: -4
A[2][2]: 0
Is it a skew-symmetric matrix ? :YES.
-- program is finished running --
```

CASE 2:

```
Please enter the size of your matrix:
2
Please enter integers in array A[] one by one:
A[0][0]: 1
A[0][1]: 2
A[1][0]: -2
A[1][1]: 1
Is it a skew-symmetric matrix ? :NO.
-- program is finished running --
```

CASE 3:

```
Please enter the size of your matrix:
3
Please enter integers in array A[] one by one:
A[0][0]: 0
A[0][1]: 1
A[0][2]: 2
A[1][0]: 1
A[1][1]: 0
A[1][2]: 3
A[2][0]: 2
A[2][1]: 3
A[2][2]: 0
Is it a skew-symmetric matrix ? :NO.
-- program is finished running --
```

CASE 4:

```
Please enter the size of your matrix:
2
Please enter integers in array A[] one by one:
A[0][0]: 1
A[0][1]: 2
A[1][0]: 3
A[1][1]: 4
Is it a skew-symmetric matrix ? :NO.
-- program is finished running --
```

CASE 5:

```
Please enter the size of your matrix:
4
Please enter integers in array A[] one by one:
A[0][0]: 0
A[0][1]: 1
A[0][2]: 2
A[0][3]: 3
A[1][0]: -1
A[1][1]: 0
A[1][2]: 2
A[1][3]: 3
A[2][0]: -2
A[2][1]: -2
A[2][2]: 0
A[2][3]: 3
A[3][0]: -3
A[3][1]: -3
A[3][2]: -3
A[3][3]: 0
Is it a skew-symmetric matrix ? :YES.
-- program is finished running --
```

Question 3: IEEE-754 Floating Point Number Decoder (20 marks)

[4 points every case]

CASE1:

```
Please enter the binary representation of your single precision
float number :
11000100001011000010001000000000
The integer part is :
-688
-- program is finished running --
```

CASE2:

```
Please enter the binary representation of your single precision
float number :
01000100001011000010001000000000
 The integer part is :
688
-- program is finished running --
```

CASE3:

```
Please enter the binary representation of your single precision
float number :
00000011100000000000000000000000
The integer part is :
0
-- program is finished running --
```

CASE4:

```
Please enter the binary representation of your single precision
float number :
11001110111111111111111111111111
The integer part is :
-2147483520
-- program is finished running --
```

CASE5:

```
Please enter the binary representation of your single precision
float number :
00111111011111111111111111111110
The integer part is :
0
-- program is finished running --
```