A. **Program with functions (15 points).** Write a working C program that will also contain the following functions: **NOTE: Don't forget to supply the function prototypes before main().**

a) **`main()`** – This function will continuously ask the user for his/her credit card purchases, will get the total of it while the user doesn't inputs 0. The function will then call the functions `amountToMiles()` and `rewards()`.

b) **`amountToMiles()`** – This function will accept as parameter total credit card purchases. It will convert the amount into miles and return miles to the calling function. (35 pesos = 1 mile)

c) **`rewards()`** – This function will accept as parameter the number of miles earned and display the following messages depending on the bracket where the number of miles will fall.

Miles	Message
Greater than 500 but less than or equal to 1,000	`You can redeem a free round trip Domestic flight.`
Greater than 1,000 but less than or equal to 3,500	`You can redeem a free round trip anywhere in Asia.`
Greater than 3,500 but less than or equal to 6,000	`You can redeem a free round trip to the Middle East.`
Greater than 6,000	`You can redeem a free round trip to America.`

B. **ARRAYS in FUNCTIONS**

Assume the ff. declaration and main() function found in the Teacher files/Inso, Mona Lisa. (array.c)

1. Make a function that inputs all components of the array.

Function Prototype:
```
void input (int a[], int size);
/*where size is the size of a[]*/
```

2. Make a function that computes the average of all the elements in the array.

Function Prototype:
```
float average (int a[], int size);
/*where size is the size of a[]*/
```

3. Make a function that returns the smallest element of the array.

Function Prototype:
```
int min (int a[], int size);
/*where size is the size of a[]*/
```

4. Make a function that will retrieve the element given the index position of the array. Return -1 if the position is invalid.

Function Prototype:
```
int retrieve (int pos,int a[],int size);
/*where pos is the index of the
element to be retrieved and size is
the size of a[]*/
```

5. Make a function that will retrieve the last element of the array.

Function Prototype:
```
int retrieve_last (int a[], int size);
/*where size is the size of a[]*/
```

6. Make a function that will retrieve the first element of the array.

Function Prototype:
```
int retrieve_first (int a[]);
```

7. Make a function that will search and retrieve the index of the element that matches integer X. Return -1 if element is not found.

Function Prototype:
```
int search (int X, int a[], int size);
/*where X is the element to be
searched and size is the size of a[]*/
```

8. Make a function that will display all the contents of the array with its corresponding index.

Function Prototype:
```
void displayAll (int a[], int size);
/*where size is the size of a[]*/
```