



Eastern Mediterranean University  
Electrical and Electronic/Information Systems/Mechatronics Engineering  
*Algorithms and Data Structures* (INFE212-EENG212)  
Mid Term Exam (Date: 18 Nov. 2017, Duration: 90 min)  
Instructor: Prof. Dr. Hasan AMCA

Question	Mark
Q1 (33 pts)	
Q2 (33 pts)	
Q3 (34 pts)	
Total points	

---

**Student Name:** \_\_\_\_\_ **Student Number:** \_\_\_\_\_

---

Important Notes  
for **Students** and **Invigilators**

1. Please Answer **All** Questions.
2. You may **use only the main course textbook** (C How to Program book by Deitel & Deitel) in the exam. No other materials are allowed.
3. Make sure that there is **no undesirable writing or note in the textbook**. If anything is written in the book, the books will be taken away and disciplinary action may follow.
4. **No mobile telephones** are allowed in the exam room.
5. No student is allowed to leave the exam room within the first **30 minutes**.
6. **No calculators** or **dictionaries** are allowed.

Q1) Use a single-subscripted array to solve the following problem. Read in 20 numbers, each of which is between 10 and 100, inclusive. As each number is read, print it only if it's not a duplicate of a number already read. Provide for the "worst case" in which all 20 numbers are different. Use the smallest possible array to solve this problem. \*/

Answer

```
/* Exercise 6.15 Solution */
```

```
#include <stdio.h>
#define MAX 20
int main()
{
    int a[ MAX ] = { 0 }; /* array for user input */
    int i; /* loop counter */
    int j; /* loop counter */
    int k = 0; /* number of values currently entered */
    int duplicate; /* flag for duplicate values */
    int value; /* current value */
```

Declare array  
(5)

Declare variables  
(5)

```
printf( "Enter 20 integers between 10 and 100:\n" );
```

```
/* get 20 integers from user */
for ( i = 0; i <= MAX - 1; i++ )
{
```

```
    duplicate = 0;
    scanf( "%d", &value );
```

```
/* test if integer is a duplicate */
for ( j = 0; j < k; j++ )
{
```

```
    /* if duplicate, raise flag and break loop */
    if ( value == a[ j ] )
    {
```

```
        duplicate = 1;
        break;
```

```
    } /* end if */
```

```
}/* end for */
```

Enter 20 integers  
(5)

Test if duplicate  
(8)

```
/* if number is not a duplicate enter it in array */
if ( !duplicate )
{
    a[ k++ ] = value;
} /* end if */

} /* end for */
```

If not duplicate  
enter in array  
(5)

```
printf( "\nThe nonduplicate values are:\n" );
```

Print non-duplicates  
(5)

```
/* display array of nonduplicates */
for ( i = 0; a[ i ] != 0; i++ )
{
    printf( "%d ", a[ i ] );
} /* end for */
```

```
printf( "\n" );
```

```
return 0; /* indicate successful termination */
} /* end main */
```

Q2) Write a complete C program, to count the number of vowels and consonants in a string using pointers. Use the following algorithm:

- Declare the related variables, integers, characters, arrays and pointers.
- Use the gets() function to read the string from the keyboard with a reminder message before entering string
- Assign the address of the read string to a pointer
- Use a while loop to check if the end of string is met. While the end of string is not met, check if the string is a vowel using a pointer to the characters of the string, that is, if it is any one of the characters a, e, i, o, u, A, E, I, O, U
- otherwise it is a consonant
- Print the total number of vowels and consonants on the screen.

Q2) Write a complete C program, to count the number of vowels and consonants in a string using pointers. Use the following algorithm:

**Answer:**

```
//Prog count vowel & consonants usng pointer
#include <stdio.h>
```

```
int main()
{
    char str[100];
    char *ptr;
    int cntV=0, cntC=0;
```

```
printf("Enter a string: ");
gets(str);
```

```
//assign address of str to ptr
ptr=str;
```

```
while(*ptr!='\0')
{
    if(*ptr=='A' ||*ptr=='E' ||*ptr=='I'
    ||*ptr=='O' ||*ptr=='U' ||*ptr=='a'
    ||*ptr=='e' ||*ptr=='i' ||*ptr=='o'
    ||*ptr=='u')
        cntV++;
```

```
else
    cntC++;
//increment pointer, to point next char
ptr++;
}
```

```
printf("Number of VOWELs: %d, CONSONANTs:
%d\n", cntV, cntC);
```

```
return 0;
}
```

Declare variables  
(5)

Use gets( ) with  
reminder msg  
(5)

Assign adress of  
string to pointer  
(5)

While loop chk '\n'  
Chk if vowel  
(8)

Otherwise a  
consonant  
(5)

Print cntV and cntC  
(5)

Q3) A dentist visits your company and asks you to prepare him a simple program to record his patients data and manage their appointments. The data required consists of ID number (6 digit integer), name (10 characters), surname (10 characters), age (2 digits), nationality (10 characters), gender (1 character), medical problem (10 characters), price to pay (in TL.kurus). When the dentist enters the ID number entered from the keyboard, all data about the particular patient will be printed on the screen. Write a complete C program to fulfill the above requirements. Use a data structure for entering data. Remember that the dentist can take care of maximum 20 patients, which will be defined as a symbolic constant. Hint: Find the array index of the patient corresponding to the ID with the algorithm: `if( patients[i].ID == IDNo ) { index=i; break; }`.

**Answer**

```
#include<stdio.h>
#define PATIENTS 2
```

Define symbolic constant  
(2)

```
struct patient
{
    int ID;
    char name[10];
    char surname[10];
    int age;
    char nationality[10];
    char gender[10];
    char problem[10];
    float price; };
```

Define structure  
(8)

```
int main()
{
    struct patient patients[PATIENTS]; // declare variables
    int i, IDNo, index;
    for(i=0; i<PATIENTS; i++)
    {
        printf("Enter id: ");
        printf("Enter name: ");
        printf("Enter Surname: ");
        printf("Enter age: ");
        printf("Enter Nationality: ");
        printf("Enter Gender: ");
        printf("Enter Problem: ");
        printf("Enter Price: ");
    }
```

Define struct variable  
(5)

Loop to enter patients  
record structure  
(5)

```
scanf("%d", &patients[i].ID);
scanf("%s", patients[i].name);
scanf("%s", patients[i].surname);
scanf("%d", &patients[i].age);
scanf("%s", patients[i].nationality);
scanf("%s", patients[i].gender);
scanf("%s", patients[i].problem);
scanf("%f.2",&patients[i].price);
```

```
printf("\n\n\nPlease enter ID number of the person");
scanf("%d", &IDNo);
```

Ask for id number  
(5)

```
for(i=0; i<PATIENTS; i++)
{
    if( patients[i].ID == IDNo )
    {
        index=i;
        break;
    }
}
```

Find index for  
given id  
(5)

Print data for  
entered id  
(5)

```
printf("The selected persons ID is = %d", patients[i].ID);
printf("\n\n\nThe data about the selected patient is : \n\n");
printf("Patients id: %d\n", patients[index].ID);
printf("Patients name: %s\n", patients[index].name);
printf("Patients Surname: %s\n", patients[index].surname);
printf("Patients age: %d\n", patients[index].age);
printf("Patients Nationality: %s\n", patients[index].nationality);
printf("Patients Gender: %s\n", patients[index].gender);
printf("Patients Problem: %s\n", patients[index].problem);
printf("Patients Price: %f.2\n", patients[index].price);
return 0;
}
```