National University of Computer & Emerging Sciences, Karachi
Spring-2017 CS-Department
Final Term Examination
22nd May 2017, 09:00 AM – 12:00 PM

Course Code:	**Course Name:**
Instructor Name / Names:	
Student Roll No:	**Section No:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **3 questions and 4 pages**.
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the sequence given in the question paper.
- This paper is subjective

Time: 180 minutes. **Max Marks** : 60 points

SECTION-I [35 minutes, 10 points (1*10)]

Question 1: Choose the best answer (Assuming that the program executes correctly):

i) What is the output of the following code:
`#include<stdio.h>`
`struct gospel{`
`int num;`
`char mess1[50];`
`char mess2[50];`
`}m1 = { 2, "if you are driven by success", "make sure that it is a quality drive"`
`};`
`void main()`
`{`
`struct gospel m2, m3;`
`m2 = m1;`
`m3 = m2;`
`printf("\n%d %s %s", m1.num, m2.mess1, m3.mess2 ) ;`
`}`
a) compile time error
b) 2 if you are driven by success make sure that it is a quality drive
c) None of the above.
	ii) Point out the correct statement which correctly free the memory pointed to by 's' and 'p' in the following program?
	`#include<stdio.h>`
	`#include<stdlib.h>`
	`int main()`
	`{`
	`struct ex`
	`{`
	`int i;`
	`float j;`
	`char *s`
	`};`
	`struct ex *p;`
	`p = (struct ex *)malloc(sizeof(struct ex));`
	`p->s = (char*)malloc(20);`
	`return 0;`
	`}`
	a) free(p); , free(p->s);
	b) free(p->s); , free(p);
	c) free(p->s);
	d) free(p);

iii)Which of the following function will be used to deallocate dynamically allocated memory ?
a) remove(var-name);
b) free(var-name);
c) delete(var-name);
d) dalloc(var-name);
	iv) Which of the following statement is correct prototype of the malloc() function in c ?
	a) int* malloc(int);
	b) char* malloc(char);
	c) unsigned int* malloc(unsigned int);
	d) void* malloc(size_t);

v) Point out the error in the following program. `#include<stdio.h>`	vi) Point out the correct statement which correctly allocates memory dynamically for 2D array following program?

```
#include<stdlib.h>

int main()
{
    char *ptr;
    *ptr = (char)malloc(30);
    strcpy(ptr, "RAM");
    printf("%s", ptr);
    free(ptr);
    return 0;
}
```

a) Error: in strcpy() statement.
b) Error: in *ptr = (char)malloc(30);
c) Error: in free(ptr);
d) No error

vii) How will you print \n on the screen?

a) printf("\n");
b) echo "\\n";
c) printf('\n');
d) printf("\\n");

Viii) Can we have an array of bit fields?

a) Yes
b) No

ix) Point out the error in the program?

```
struct emp
{
    int ecode;
    struct emp e;
};
```

a) Error: in structure declaration
b) Linker Error
c) No Error
d) None of above

x) What will be the output of the program ?

```
#include<stdio.h>
#include<string.h>

int main()
{
char str1[20] = "Hello", str2[20]=" World";
printf("%s\n", strcpy(str2, strcat(str1, str2)));
    return 0;
}
```

a) Hello
b) World
c) Hello World
d) WorldHello

SECTION-II [45 minutes, 10 points (2*10)]

Question 2: In this question, provide **1-2** line explanation. Where appropriate, you may use a short fragment of code to complement your explanation. **N.B: Negative marking on more than 2 lines.**

(i): What is the difference between a local and global variable in C? (Consider variable scope, storage and initialization).

(ii): What are the properties of a static member variable in C?

(iii): Briefly explain pointer arithmetic in C. Give an example code snippet involving pointers in which it would be inappropriate to use pointer arithmetic, and explain why.

(iv): Explain how in some respect pointers are equivalent to arrays, and give one respect in which they differ.

(v): What is the difference between declaration and definition?

(vi): Describe the layout of the memory components: Dynamic Memory Allocation, Data Segment, Code Segment and Stack. You may use an illustration as part of your explanation.

(vii): List some of the ways recursion can improve your applications?

(viii): Compare and contrast structure and union.

(ix): Explain the three stages of compilation in C language.

(x): Write down the benefit of **typedef** with one example.

SECTION-III [100 minutes, 30 points (6*5)]

Question 3: This section comprises programs.

a) Write a program, which stores information about a date in a structure containing three members i.e. day, month and year. Using bit fields the day number should get stored in first 5 bits of day, the month number in 4 bits of month and year in 12 bits of year. Write a program to read date of joining of 10 employees and display them in ascending order of year.

b) Suppose a file contains a multiple line text. Write a program to replace first letter of every word with caps in a same file. Sample input/output file is given below.

Hint: Use fseek() function to position your pointer before over writing the letter in caps.

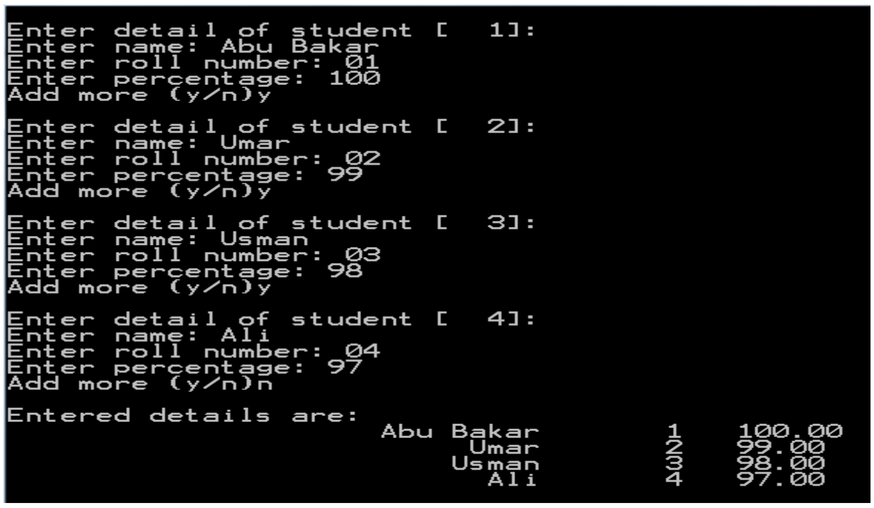
Sample Input/Output:

MyFile.txt (Input)	MyFile.txt (Output)
don't be stressed do your best forget the rest	Don't Be Stressed Do Your Best Forget The Rest

c) Write a program that will create a structure with N number of student details (Name, Roll# and Percentage) and print the inputted details. Memory to store and print structure will be allocated at run time by using malloc() and released by free().

Hint: N number of students should be managed using realloc() function.

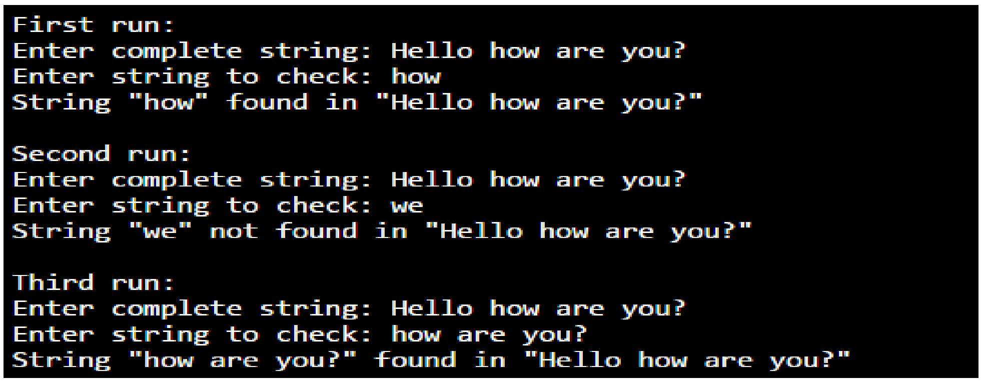
Sample Input/Output:



```
Enter detail of student [   1]:
Enter name: Abu Bakar
Enter roll number: 01
Enter percentage: 100
Add more (y/n)y

Enter detail of student [   2]:
Enter name: Umar
Enter roll number: 02
Enter percentage: 99
Add more (y/n)y

Enter detail of student [   3]:
Enter name: Usman
Enter roll number: 03
Enter percentage: 98
Add more (y/n)y

Enter detail of student [   4]:
Enter name: Ali
Enter roll number: 04
Enter percentage: 97
Add more (y/n)n

Entered details are:
                     Abu Bakar         1   100.00
                          Umar         2   99.00
                         Usman         3   98.00
                           Ali         4   97.00
```

d) Write a program to check whether a substring is present in a string or not without using library function. For this you have to implement a function named ***myStrStr()***, this function will return 1 if substring present in the string and return 0, if substring is not present in the string.

Sample Input/Output:



```
First run:
Enter complete string: Hello how are you?
Enter string to check: how
String "how" found in "Hello how are you?"

Second run:
Enter complete string: Hello how are you?
Enter string to check: we
String "we" not found in "Hello how are you?"

Third run:
Enter complete string: Hello how are you?
Enter string to check: how are you?
String "how are you?" found in "Hello how are you?"
```

e) In numerical analysis, a sparse matrix is a matrix in which most of the elements are zero. By contrast, if most of the elements are nonzero, then the matrix is considered dense. The fraction of zero elements (non-zero elements) in a matrix is called the sparsity (density).

Write down a C program that takes a matrix of 5X5 dimensions and finds whether this matrix is sparse or dense, and displays message accordingly. [Only write core logic, no need for # include]

BEST OF LUCK!