PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

END SEMESTER ASSESSMENT (ESA) - JULY - 2023

UE22CS151B - Problem Solving With C

Total Marks : 100.0

1.a. Draw a neat picture and explain each step in the Program Development Life Cycle (**PDLC**) of a C program. (6.0 Marks)

1.b. **Mention the output of the following code pieces individually.**

i) int n=7980; printf("%d", printf("%d", printf("%d", n)));
ii) int case = 8; printf(“%d”,case);
iii) printf("%d", -4?0:4);
iv) int a; printf("%d",a = 2 | (7 == 7 == 7)); (4.0 Marks)

1.c. Write a C Program to count the number of digits in a number taken through user input and also check if the individual digits are even or odd.
Display appropriate messages.

Sample output:

```
Enter a number: 12345
5 is odd
4 is even
3 is odd
2 is even
1 is odd
The number of digits in the number is :5
```

(5.0 Marks)

1.d. i) **State True or False :**
a) ++ is a unary operator in C.
b) C is an interpreted language.
c) There can be multiple a.exe files in one folder.

ii) How many bytes does **sizeof(‘\r’)** occupy?

iii) **Scanf()** instead of **scanf()** leads to ________ error. (5.0 Marks)

2.a. Write a C function **my_strcat()** that accepts two strings as arguments, which are taken as user input, and emulates **strcat()** in the **string.h** file.
Test this function with the client code.

Sample Output:

```
Enter str1:
Exam
Enter str2:
Over
str1 is ExamOver and str2 is Over
```

(6.0 Marks)

2.b. **Find the output of the following program.**

```
#include<stdio.h>
int main()
{
char str[] = "PESU";
int i;
for(i=0; str[i]; i++)
printf("%c %c %c %c\n", str[i], *(str+i), *(i+str), i[str]+2);
}
```

(4.0 Marks)

2.c. i)**Define Recursion?**

ii**) Find the output of the following C code**.

```
#include<stdio.h>
int what(int num,int res);
int main()
{
int a = 151;
printf("%d\n",what(a,0));
return 0;
}

int what(int num,int res)
{
if(num==0)
return res;
else
return what(num/10,res+(num%10));
}
```

(4.0 Marks)

2.d. Write a C function that returns the smallest element from an integer array **arr** with **n** elements. In the main function, call the function to the test.

For Example: If the array elements are **{9,7,5,3,10,12,5},** the function returns the smallest element of the array, which is 3 in this case. (6.0 Marks)

3.a. i) **Find the output of the below C program.   // 2 marks**

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int *p1 = (int*)malloc(sizeof(int));
*p1 = 250;
printf("%d ", *p1);
int *p2 = p1;
printf(" %d ", *p2);
*p2 = 999;
printf("%d ",*p1);
printf("%d ",*p2);
free(p1);
p1 = NULL;
p2 = NULL;
}
```

ii) If the same pointer variable is allocated memory more than once using the dynamic memory allocation functions, initially allocated memory space becomes a _________.
Garbage which has no name and hence no access in turn results in __________   **// 2 marks**

(4.0 Marks)

3.b. **The below C program has errors. Recognise the errors and write a correct program to produce the output 1001 and XYZ separated by a tab space.**

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

struct Student
{
int roll_no;
char name[100];
};
int main()
{
struct Student s;
s = malloc(1,sizeof(struct Student));
s.roll_no = 1001;
s->name = "XYZ";
printf("%d\t%s\n",s.roll_no,s.name);
return 0;
}
```

(5.0 Marks)

3.c. In ABC company there are 3 salesmen. Each salesman sells 2 items.
Write a C program using **two dimensional arrays** to display the **total sales of each item.**

Sample Output:

```
Enter the data:
Enter the sales of 2 items sold by the sales man: 0
23   45
Enter the sales of 2 items sold by the sales man: 1
20   40
Enter the sales of 2 items sold by the sales man: 2
10   15
Total sales of each item 0 = 53
Total sales of each item 1 = 100
```

(5.0 Marks)

3.d. Given the structure declaration and the client code, define the function **insert_front** to add nodes to the beginning of the linked list and define the **display** function as well to print the data in the nodes.

Sample Output:

```
Enter the element:
10
Enter the element:
20
Enter the element:
30
30 20 10
```

```
typedef struct node
{
   int data;
   struct node *link;

}node;

int main()
{
   int element;
   node *head = NULL;
   for(int i=0;i<3;i++)
   {
   printf("Enter the element:\n");
   scanf("%d",&element);
   head = insert_front(head,element);
   }
   display(head);
}
```

(6.0 Marks)

4.a. The file **test.txt** exists with some data. Write a C Program to find the length of the text file using **fseek()** and **ftell()** functions. (5.0 Marks)

4.b. Given a sorted array of integers, write a function which searches for a given integer using binary search and returns the index of it, returns -1 otherwise.
Use the below function declaration to define the function.

int binary_search(int *array, int size, int value);

array: pointer to the array where the value has to be searched.
value : element to be searched
size : size of the array (6.0 Marks)

4.c. **i) Find the output of the below code.**

```
 #include <stdio.h>
int main()
{
int arr[6] = {10,7,11,18};
int *arrp[10];
arrp[9] = &arr[3];
printf("%d ",*arrp[9]);
arrp[11] = &arr[5];
printf("%d\n",*arrp[11]);
}
```

ii) **Find the output of the following code**

```
#include <stdio.h>
#include<string.h>
int main()
{
char line[100]="PESU,EC,RR";
printf("%s ",strtok(line,","));
printf("%s ",strtok(NULL,","));
printf("%s ",strtok(line,","));
}
```

(4.0 Marks)

4.d. Define callbacks in C with an example program. **(2 marks -Definition, 3 marks - example program)** (5.0 Marks)

5.a. Give brief notes on the following keywords with suitable code snippets:
i) **volatile**
ii) **extern** (4.0 Marks)

5.b. **Find the output of the following program.**

```
#include <stdio.h>
enum marks{Phy=1,Chem=4,Maths,Comp};
int main()
{
enum marks m;
m=Maths;
printf("%d ",Chem);
switch(m)
{
case Phy:printf("Physics");break;
case Chem:printf("Chemistry");break;
case Maths:printf("Maths");break;
case Comp:printf("Computers");break;
}
printf(" %d ",Comp);
printf("%d ",Phy);
}
```

(4.0 Marks)

5.c. **Find the output of the following C programs.**

i)

```
#include<stdio.h>
int main()
{     char a = 'z';            char b = 'w';          const char *c = &b;          *c = 'x';
printf("%c",*c);              return 0;
}
```

ii)

```
#include<stdio.h>
int main()
{  int i = 333;         int j = 666;         int* const p = &i;      *p = 555;
  printf("%d\n",*p);          return 0;}
```

iii)

```
#include<stdio.h>
int main()
{    printf("%d",sizeof(long) >= sizeof(int));         return 0;       }
```

iv)

```
#include<stdio.h>
#include<stddef.h>
union A
{   int x;      float y;        char z;   };
int main()
{       printf("%lu ",offsetof(union A,z));
}
```

v)

```
#include<stdio.h>
void fun();
int main()
{        fun();        fun();        return 0; }
void fun()
{        static int a = -1;        a--;        printf("%d\t",a);        }
```

(6.0 Marks)

5.d. **i) Mention any 3 differences between unions and structures**

ii) Find the output of the following code.

```
#include<stdio.h>
#define MAX 5
#define fun(a,b) a*b
int main()
{
   printf("%d\t",fun(MAX,3+6));
   #undef MAX
   int MAX = 35;
   printf("%d\t",MAX);
   #define MAX 22
   printf("%d",MAX);
}
```

(6.0 Marks)