Paper Code - U239-145 (T1)

OCTOBER 2019 / INSEM (T1)
S. Y. B.TECH. (INFORMATION TECHNOLOGY) (SEMESTER – III)
COURSE NAME: FUNDAMENTALS OF DATA STRUCTURES
COURSE CODE: ITUA21185
(PATTERN 2018)
SOLUTION

Q 1) a) 1.

```
void xstrcat(char *name1,char *name2)
{
int i=0,len;
char *p;
p=name1;
xstrcpy(p,name1);
len=xstrlen(name1);
while(*(name2)!='\0')
    {
      *(p+len)=*name2;
    len++;
    name2++;
}
*(p+len)='\0';
printf("%s",p);
}
```

2.

- Pointers allow you to refer to large data structures in a compact way

Eg- array, database (array of structure elements)

- Pointers facilitate sharing between different parts of programs using Call by reference in functions
- Pointes make it possible to get new memory dynamically as your program is running i.e. Dynamic memory allocation using malloc, calloc, realloc
- They make it easy to represent relationships among data items. Complex data structures like linked list, stack, queue, tree, graph etc.

Q-2 a) 1.

```
void main(int argc, char* argv[])
{
FILE *fs,*ft;
char ch;
if(argc!=3)
        {
        printf("Incorrect Number of Command Line Arguments");
        return;
        }
fs=fopen(argv[1],"r");
ft=fopen(argv[2],"w");
if(fs==NULL || ft==NULL)
        {
        printf("cannot open files!!");
        return;
```

```
        }
while(1)
        { ch=fgetc(fs);
        putc(ch,ft);
        if(ch==EOF) break;
        }
        fclose(fs);
        fclose(ft);
}
```

2. Pointer to function/ function pointers:
    - Contains address of function
    - Similar to how array name is address of first element
    - Function name is starting address of code that defines function

Function pointers can be :
    - Passed to functions
    - Stored in arrays
    - Assigned to other function pointers

Q 2) a) 1. ADTs are one way of separating parts of a larger programming task from the rest of the program.

- ☐ This allows for better structuring, and it enables multiple programmers to work on the same project.
- ☐ When done wisely it can be used to build libraries with code that can be incorporated into many *different* projects.

ADT for a Matrix: Data: number of rows, number of columns, datatype
Operations: init( ), add( ),mult( ), transpose( )
2. total cost equation: 3M time complexity : 1M O(n^3)

b) 1. Big-oh notation Definition: 1.5 M, Graph:1.5M, Example: 1M
2. Differentiation (Min 2 points)
    i) Static and dynamic data structures: 2M
    ii) Linear and non-linear data structures: 2M

Q 3) a) Position=binarysearch(data,key,0,n-1); //call

```
int binarysearch(int s[20],int key,int low,int high)//recursive definition
        {   int mid;
        if(low>high)  return (-1);
        mid=(low+high)/2;
        if(key=s[mid]) return(mid);
        if(key>s[mid])  return(binarysearch(s, key, mid+1,high));
        return(binarysearch(s ,key, low, mid-1));
        }
```

b) Pass-wise insertion sort: 56, 12, 54, 28, -13, 47, 94, -2.

Pass1: **56**, 12, 54, 28, -13, 47, 94, -2.
Pass2: **56, 12**, 54, 28, -13, 47, 94, -2.
Pass3: **56, 54,12,** 28, -13, 47, 94, -2.
Pass4: **56, 54,28,12,**-13, 47, 94, -2.
Pass5: **56, 54,28,12,-13**, 47, 94, -2.
Pass6: **56, 54,47,28,12,-13,** 94, -2.
Pass7: **94,56, 54,47,28,12,-13,**-2.