13.5/15

Quiz 3: Set A

Introduction to Computing and Programming (CSD101)

Max. Marks: 15 **Date: 21-11-2024**

Duration: 30 min.

Name: ADYA SINGHAL **Roll No.** 2410110027

Q.1 struct node

```
{
  int i;
  float j;
};
struct node *s[10];
```

The above C declaration define 's' to be **(2 Marks)**

[✓ a)] An array, each element of which is a pointer to a structure of type node

b) A structure of 2 fields, each field being a pointer to an array of 10 elements

c) A structure of 3 fields: an integer, a float, and an array of 10 elements

d) An array, each element of which is a structure of type node.

2

Solution: *s[10] means an array of 10 elements where each is a pointer (stores address) to a structure of type node.

Q.2. What does the following statement mean? **Provide justification for your answer. (2 marks)**

int (*fp)(char*)

a. pointer to pointer

b. pointer to an array of chars

[✓ c.] pointer to function taking a char* argument and returns an int

d. More than one of the above

e. None of the above

Solution:

2

This means :

$$\underbrace{\text{int}}_{\text{return type of function}}\ \underbrace{(*fp)}_{\text{name of function pointer}}\ \underbrace{(\text{char}*)}_{\text{function argument type}}$$

Q.3. What is the use of function char *strchr(ch, c)? **Provide one line justification for your answer** **(1 mark)**

a) return pointer to first occurrence of ch in c or NULL if not present

[✓ b)] return pointer to first occurrence of c in ch or NULL if not present

c) return pointer to first occurrence of ch in c or ignores if not present

d) return pointer to first occurrence of cin ch or ignores if not present

Solution:

strchr returns pointer to first occurence.

Eg. strchr (str, char 1) → char 1 is the character whose ~~locat~~ position we can find.

It will return pointer to first occurence of char 1 in str, if char 1 is present, else it will return NULL pointer.

1

Q.4. **What is the output of the following program? Provide reasoning for the same. (2 marks)**

```
#include <stdio.h>

int main() {

double a[3]={20.0,25.0,99.0},* p,* q;

p=a+1;

q=p++ ;

printf("%d,%d", (int) (q-p),( int)(* q-* p));

return 0;}
```

Solution:

2

OUTPUT:- 8 74

we get this output as:- (a points to 20.0)

p=a+1 i.e. p points to 25.0 ⇒ *p=25.0

q=p++ i.e. q points to 90.0 ⇒ *q=99.0

1) ∴ (int)(q-p) will give sizeof(double) ∵ elements of array are of double type ⇒ sizeof(double)=8

2) (int)(*q-*p) will give 99-25=74.

Q.5. Write 2 difference between calloc() and malloc()? Write the syntax for both calloc() and malloc(). **(4 marks)**

Solution:

Calloc ()	Malloc ()
→ Allocates contiguous memory locations of specified size and number.	→ Allocates a single large block of memory of specified size.
→ Takes 2 arguments / parameters.	→ Takes 1 arguement / parameter
SYNTAX:	SYNTAX:
int *ptr = (int*) calloc (n, sizeof(int));	int* ptr = (int*) malloc (10*sizeof (int));
∵ sizeof(int) = 4, this allocates n blocks of size 4.	∵ sizeof(int) = 4, this allocates a single block of memory of size = 10×4 = 40.

4

Q.6 Illustrate the working of selection sort with example. **(4 marks)**

Solution:

~~ROUGH:~~
~~{1,3,2,4}~~
~~min= 0~~
~~min= 1~~

```
void sel_sort (int a[], int n){
   for (int i=0; i<n; i++){
        int min-idx = i;

        for (int j=i+1; j<n; j++){
             if (a[j] > a[j+1]) {
                  min- idx = j; }
        }
        int temp = a[i];
        a [i] = a[min -idx];
        a [min- idx] = temp;
   }
}

int main () {
   int arr1[] = {10, 40, 50, 30};
   int n = sizeof(arr)/sizeof(arr[0]);
   sel sort (arr1, n);
   for (int K=0; K<n; K++){              // to print the sorted array
       printf("%d", arr1[K]);}
}
```

4