

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.

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:

This means:

int (*fp)(char*)
↓ ↓ →
return type of name of function argument type
function function pointer

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` in `ch` or ignores if not present

Solution:

`strchr` returns pointer to first occurrence.

Eg. `strchr(str, char1)` → `char1` is the character whose ~~total~~ position we can find. It will return pointer to first occurrence of `char1` in `str`, if `char1` is present, else it will return `NULL` pointer.

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:

OUTPUT: 8 74

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

`p=a+1` i.e. `p` points to 25.0 $\Rightarrow *p=25.0$

`q=p++` i.e. `q` points to 99.0 $\Rightarrow *q=99.0$

1) $\therefore (int)(q-p)$ will give `sizeof(double)` \because elements of array are of double type $\Rightarrow \text{sizeof(double)}=8$

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()

→ Allocates contiguous memory locations of specified size and number.

→ Takes 2 arguments / parameters.

SYNTAX:

$\text{int}^* \text{ptr} = (\text{int}^*) \text{calloc}(n, \text{sizeof}(\text{int}));$

∵ $\text{sizeof}(\text{int}) = 4$, this allocates n blocks of size 4.

Malloc()

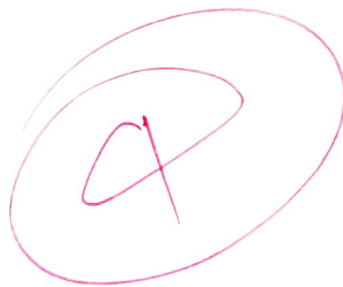
→ Allocates a single large block of memory of specified size.

→ Takes 1 argument / parameter

SYNTAX:

$\text{int}^* \text{ptr} = (\text{int}^*) \text{malloc}(10 * \text{sizeof}(\text{int}));$

∵ $\text{sizeof}(\text{int}) = 4$, this allocates a single block of memory of size $= 10 \times 4 = 40$.



Q.6 Illustrate the working of selection sort with example.

(4 marks)

Solution:

```
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[min-idx]) {  
                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(arr1)/sizeof(arr1[0]);  
    sel-sort(arr1, n);  
    for (int k=0; k<n; k++) {  
        printf("%d", arr1[k]);  
    }  
}
```

// to print the sorted array

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