Summary in Graph

Exam Summary (GO Classes Test Series 2024 | Programming | Test 3)

Qs. Attempted:	**12** 5 + 7	**Correct Marks:**	**11** 1 + 10
Correct Attempts:	**6** 1 + 5	**Penalty Marks:**	**1.67** 0.33 + 1.33
Incorrect Attempts:	**6** 4 + 2	**Resultant Marks:**	**9.33** 0.66 + 8.66

Total Questions:	**15** 5 + 10
Total Marks:	**25** 5 + 20
Exam Duration:	**45 Minutes**
Time Taken:	**45 Minutes**

EXAM RESPONSE | EXAM STATS | FEEDBACK

Technical

Q #1 | Multiple Select Type | Award: 1 | Penalty: 0 | Programming in C

In which of the following case(s) character array must end with null char?

A.
```
char c[] = "abcd";
```

B.
```
char c[] = {'a', 'b', 'c', 'd'};
```

C.
```
char c[5] = "abcd";
```

D.
```
char c[4] = "abcd";
```

Your Answer: A;B;D | Correct Answer: A;C | Incorrect | Discuss

Q #2 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | Programming in C

What will be the output of a given program in a little endian system?

```
#include
int main()
{
    char t[] = "abcdefg";
    int *p = t;
    p++;
    char *c = p;
    printf("%c", *c);
}
```

A. b
B. d
C. e
D. f

Your Answer: A **Correct Answer: C** Incorrect Discuss

Q #3 Multiple Select Type Award: 1 Penalty: 0 Programming in C

Which of the following may lead to a runtime error for the given declaration of p?

```
int (*p)[5];
```

A. `p[1][1]`
B. `*p[1]`
C. `(*p)[1]`
D. `p[1]`

Your Answer: A;B;D **Correct Answer: A;B** Incorrect Discuss

Q #4 Multiple Choice Type Award: 1 Penalty: 0.33 Programming in C

Consider the following C program.

```
#include<stdio.h>
void fun (int a[5]){
    for (int i =0; i<8; i++)
        a[i] = a[i]+1;
}
main ()
{
    int a[8] = {1,2,3,4,5,6,7,8};
    fun(a);
    for (int i =0; i<8; i++)
    printf("%d", a[i]);
}
```

Which of the following is/are true about the given program?

A. Program will exhibit a compiler error since array “a” is of size 5 in function fun() but for loop in fun() runs till 8.
B. Program will exhibit run time error since array “a” is of size 5 in function fun() but for loop in fun() runs till 8.
C. On running the program, the output will be 23456789
D. None of the above

Your Answer: C | Correct Answer: C | Correct | Discuss

Q #5 | Multiple Select Type | Award: 1 | Penalty: 0 | Programming in C

Which of the following is/are valid assignment(s) of the pointer for the given declaration. An assignment is valid if both pointers are of the same type (pointers are compatible) and get successfully compiled with no warning or error.

```
int a[4][4];
int *t, **pp;
```

A.
```
pp = a;
```

B.
```
pp = &a;
```

C.
```
pp = *a;
```

D.
```
t = a[2];
```

Your Answer: A;D | Correct Answer: D | Incorrect | Discuss

Q #6 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

Consider the given declaration

```
int arr[3][4];
int (*p)[12];
p = malloc(sizeof(int) * 24); // assume malloc() is successful
```

```
sizeof(int) = 4 bytes and sizeof(int *) = 8 bytes.
```

What will be the output of the following lines of C code ?

```
printf("%d " , sizeof(arr[2]));
printf("%d ", sizeof(p));
printf("%d", sizeof(*p));
```

A. 8 8 8
B. 8 8 48
C. 16 8 8
D. 16 8 48

Your Answer: D | Correct Answer: D | Correct | Discuss

Q #7 | Multiple Select Type | Award: 2 | Penalty: 0 | Programming in C

The C code printed below is a version of the C string library call strlen(), which returns the length of a given string.

```
unsigned int mystrlen(char *c)
{
    unsigned int i =0;
    /* Your code here. */
5.  return i;
}
```

Which line of code creates a working and accurate version of strlen()?

A.
```
while(*c!='\0'){
    i++;
    c++;
  }
```

B.
```
while(*c++!='\0') i++;
```

C.
```
while(*c++ && i++);
```

D.
```
while (c[i++] != '\0');
```

Your Answer: A;B | Correct Answer: A;B | Correct | Discuss

Q #8 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

What will be the output of the following lines of C code?

```
int a[] = {1,2,3,4};
int *p = (int *)(&a+1);
printf("%d", *(p-1));
```

A. 1
B. 2
C. 3
D. 4

Your Answer: D | Correct Answer: D | Correct | Discuss

Q #9 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

What will be the output of the following C code?

```
#include<stdio.h>
void strXpos(char * dest, char * src) {
    if (src[0] == '\0') {
        dest[0] = '\0';
        return;
    }
    if (src[1] == '\0') {
        dest[0] = src[0];
        dest[1] = src[1];
        return;
    }
    dest[0] = src[1];
    dest[1] = src[0];
    strXpos(dest + 2, src + 2);
}
int main() {
    char s1[10] = "GOGATE";
    char s2[10] = "GO";
    strXpos(s2, s1);
    printf("%s", s2);
}
```

A. GOGATE
B. OGAG
C. OGAGET
D. GOAGET

Your Answer: | Correct Answer: C | Not Attempted | Discuss

Q #10

Numerical Type | Award: 2 | Penalty: 0 | Programming in C

What is the maximum value of i that will not cause this code to overflow?

It is given that sizeof(int) = 4 and sizeof(int *) = 8.

```
int *p = malloc(sizeof(int *) * 2);
p[i] = 2501;
```

Your Answer: 3 | Correct Answer: 3 | Correct | Discuss

Q #11

Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

What will be the output of the following program?

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

struct _info {
    char **strs;
};
typedef struct _info info;
main(){
    info arr_str[2];
    char *strs1[] = {"GO", "Classes"};
    char *strs2[] = {"GATE", "2023"};
    arr_str[0].strs =strs1;
    arr_str[1].strs =strs2;
    info **da, *a;
    a =arr_str;
    da = &a;
    for (int i = 0; i<2; i++)
        for (int j = 0; j< 2; j++){
            printf("%s ", (((*da)+i)->strs)[j]);
        }
}
```

A. GO GATE Classes 2023
B. GO Classes GATE 2023
C. GATE GO 2023 Classes
D. GATE 2023 GO Classes

Your Answer: | Correct Answer: B | Not Attempted | Discuss

Q #12

Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

What will be the output of the program below-

```
    int u[] = {1, 7, 8, 3};
    int v[] = {-4, 9, 33};
    int w[] = {12, 2, 57};

 5. struct t{
        char *c;
        int *t;
    }go[] = {"GATE", u, "GOClasses", v, "GATEOVERFLOW", w};

10. main()
    {
        struct t *p = go;
        printf("%s ", ++p -> c);
        printf("%d ", ++*p -> t);
15.     printf("%d ", *++p[0].t);
        printf("%s ", (++p) -> c);
    }
```

A. ATE 1 9 GATEOVERFLOW
B. GATE -4 2 GATEOVERFLOW
C. ATE 2 7 GOClasses
D. None of the above

Your Answer: D **Correct Answer: C** Incorrect Discuss

Q #13 Multiple Choice Type Award: 2 Penalty: 0.67 Programming in C

Assume that variables $p1, p2$ and n are defined as follows:

```
    int *p1 = (int *) malloc(2 * sizeof(int));
    int *p2 = p1 + 1;
    int n = 1;
```

Given 4 programs and 4 types of issues below, Check which program is likely to cause which type of issue.

Prog1:
free(p1);
*(p2 - 1) = n;
free(p2);
	A. Using unallocated memory

Prog2:
free(p1);
p2 -= 1;
	B. Memory leak

Prog3: *(p2 + 1) = n; free(p1);	C. Dangling Pointer

Prog4:
p1 = &n;
*p1 = n;
free(p1);
	D. No Issue

A. Prog1-C, Prog2-C, Prog3-A, Prog4-D
B. Prog1-C, Prog2-D, Prog3-A, Prog4-B
C. Prog1-C, Prog2-D, Prog3-A, Prog4-D
D. Prog1-D, Prog2-C, Prog3-A, Prog4-B

Your Answer: B | Correct Answer: B | Correct | Discuss

Q #14 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

Consider struct alignment rules as described below –

1. Each primitive data (int, char, float etc) type requiring K bytes must be stored at the address must be multiple of K.
2. Address of structure and structure length must be multiples of $\mathrm{K}_{\max}$. Where $\mathrm{K}_{\max}$ is the largest size of any primitive element of the struct.

Also, we define internal and external fragmentation of struct as follows–

- Internal fragmentation is padding used between any two primitive data types of the struct.
- External fragmentation is padding used at the end of the last element of struct to fulfill 2^{nd} rule requirement of alignment above.

For a given struct below, what will be internal and external fragmentation, respectively? Consider that the system has 8 bytes of address lines, 4 bytes of integers, 2 bytes of short, and 1 byte of the character.

```
typedef struct {
    int num_donors;
    char** donor_names;
    char has_soft_seats;
    char location_name[10];
    short height;
} cse_building;
```

A. $4, 3$
B. $5, 2$
C. $0, 7$
D. $4, 0$

Your Answer: | Correct Answer: B | Not Attempted | Discuss

Q #15 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Programming in C

Consider the following declarations of variables in a system having 4 bytes for integers.

```
int a[4][4];
int (*p1)[2];
int (*p2)[2];
int (*t1)[4];
int (*t2)[4];
```

Suppose array a has base address of 1000 and initial values of $p1, p2, t1$, and $t2$ are $1040, 1064, 1044$ and 1076 respectively.

What will be the value of $p2 - p1$ and $t2 - t1$ respectively?

A. $6, 8$
B. $3, 2$
C. $12, 8$
D. $32, 24$

Your Answer: C | Correct Answer: B | Incorrect | Discuss
