Summary in Graph

Exam Summary (GO Classes CS Test Series | C-Programming | Topic Wise Test 3)

38.7% Score

6 Correct | 6 Wrong | 3 Skipped | 44m 4l Min

Qs. Attempted:	12 of 15	Correct Marks:	11	Total Questions:	15
Correct Attempts:	6 of 6	Penalty Marks:	1.33 -0.33	Total Marks:	25 +1.00
Incorrect Attempts:	6 -0.33	Resultant Marks:	9.67 of 25.00	Exam Duration:	45 Minutes
				Time Taken:	44 Minutes

Individual Question Statistics

View Order: Sorted Order | Select Metric: Marks Obtained



* Average Stats based on first attempt stats of Top 10% (100) out of 1001 users

Hide Incorrect | Hide Info | Reports

Technical

Q.11 Multiple Select Type | Point 2 | Penalty 0 | Programming in C | Medium (40%)

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

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

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

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

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

Your Answer: A,B,C Correct Answer: A,C Incorrect Time Taken: 00m 00s Skipped

Q.12 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Hard (24%)

What will be the output of a given program in a little endian system? (Assume int takes 4 bytes)

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

A. 0
B. d
C. e
D. f

Your Answer: C Correct Answer: C Not Attempted Time Taken: 00m 00s Skipped

Q.13 Multiple Select Type | Point 2 | Penalty 0 | Programming in C | Hard (21%)

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 Correct Answer: A,B Incorrect Time Taken: 00m 00s Skipped

Q.14 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Hard (28%)

Consider the following C program.

```
#include<stdio.h>
void fun (int a[])
{
    for (int i = 0; i < 5; i++)
        a[i] = a[i] * a[i];
}
main ()
{
    int a[5] = {1,2,3,4,5};
    fun(a);
    for (int i =0; i<5; 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 1916457169
D. None of the above

Your Answer: C Correct Answer: C Correct Time Taken: 00m 00s Skipped

Q.15 Multiple Select Type | Point 2 | Penalty 0 | Programming in C | Hard (15%)

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[2][4];
int *b,**pp;
```

A. `pp = a;`

B. `pp = &a;`

C. `pp = *a;`

D. `b = A[0];`

Your Answer: A,B,D Correct Answer: B Incorrect Time Taken: 00m 00s Skipped

Q.16 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Hard (4%)

Consider the given declaration

```
int arr[2][3];
int (*ptr)[3];
p = malloc(sizeof(int) * 20); // 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));
printf("%d ", sizeof(ptr));
printf("%d", sizeof(*ptr));
```

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

Your Answer: D Correct Answer: D Correct Time Taken: 00m 00s Skipped

Q.17 Multiple Select Type | Point 2 | Penalty 0.33 | Programming in C | Hard (26%)

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 *s)
{
    unsigned int n =0;
    /* User code here. */
    return n;
}
```

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

A.
```
while(*s++ != '\0')
    n++;
```

B.
```
while(*s++ != 0)
    ;
```

C.
```
while(*s != 0) n++;
```

D.
```
while (s[n++] != '\0');
```

Your Answer: A,B,D Correct Answer: A,B Incorrect Time Taken: 00m 00s Skipped

Q.18 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Medium (34%)

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

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

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

Your Answer: B Correct Answer: B Correct Time Taken: 00m 00s Skipped

Q.19 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Easy (8%)

What will be the output of the following C code?

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

A. GOGATE
B. OGAG
C. OGAGGT
D. GOAGGT

Your Answer: C Correct Answer: C Correct Time Taken: 00m 00s Skipped

Q.20 Numerical Type | Point 2 | Penalty 0.33 | Programming in C | Hard (24%)

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 *) * i);
a[i] = 100;
```

Your Answer: 0 Correct Answer: 0 Correct Time Taken: 00m 00s Skipped

Q.21 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Medium (34%)

What will be the output of the following program?

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

struct _info {
    char *vers;
};
typedef struct _info info;
Author{
    info arr_str[2];
} Author[2] = {"GO","Classes"};
char *strval = "TEST", *str2;
int intval = 2020, intval2;
arr_str[1].vers =&intval;
info **ik, *ik;
ik = &ik;
ik = &ik;
for (int i = 0; i<2; i++)
    for (int j = 0; j< 2; j++)
        printf("%s ", ((*ik)+i)->vers[j]);
}
```

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

Your Answer: B Correct Answer: B Correct Time Taken: 01m 00s Skipped

Q.22 Multiple Choice Type | Point 2 | Penalty 0.33 | Programming in C | Medium (40%)

What will be the output of the program below:

```
int a[ ] = {1, 3, 5, 7};
int b[ ] = {4, 6, 7, 9};
int m[ ] = {11, 2, 17};

struct X{
    char *s;
    int *t;
}p[] = {"GATE", a, "GOClasses", a, "GATEWORKSHOP", m};

main()
{
    struct X *q = p;
    printf("%c ", ++*q->s);
    printf("%d ", *++q->t);
    printf("%s ", ++q->s);
    printf("%d ", (++q)->t);
}
```

A. 67L 3 G ATEWORKSHOP
B. 67L 3 GATEWORKSHOP
C. H3 3 ITGOClasses
D. None of the above

Your Answer: B Correct Answer: B Incorrect Time Taken: 00m 00s Skipped

Q.23 Multiple Choice Type | Point 2 | Penalty 0.67 | Programming in C | Hard (27%)

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

```
int *p1 = (int *) malloc(1 * sizeof(int));
int *p2 = p1 + 1;
int a = 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) = a; free(p2);`	A. Using unallocated memory
Prog2 : `free(p1); p2 += 1;`	B. Memory leak
Prog3 : `*(p2 + 1) = a; free(p1);`	C. Dangling Pointer
Prog4 : `p1 = \&a; *p1 = a; 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-B, Prog2-C, Prog3-A, Prog4-B

Your Answer: Correct Answer: B Not Attempted Time Taken: 00m 00s Skipped

Q.24 Multiple Choice Type | Point 2 | Penalty 0.67 | Programming in C | Medium (34%)

Consider struct alignment rules as described below –

1. Each primitive data (int, char, float etc) type requiring N bytes must be stored at the address must be multiple of N.
2. Address of structure and structure length must be multiples of K_{max} Where 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 2nd 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_doors;
    char* owner_name;
    int* last_service_dates;
    char location_name[10];
    short height;
} car_building;
```

A. 6, 3
B. 5, 2
C. 3, 7
D. 4, 0

Your Answer: Correct Answer: B Not Attempted Time Taken: 00m 00s Skipped

Q.25 Multiple Choice Type | Point 2 | Penalty 0.67 | Programming in C | Medium (34%)

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

```
int a[3][5];
int (*ptr)[5];
int (*ptr1)[];
int **ptr2;
int *ptr[4];
```

Suppose array a has base address of 1000 and initial values of p1, p2, r1, and r2 are 1040, 1064, 1064 and 1076 respectively.

What will be the value of p2 - p1 and r2 - r1 respectively?

A. 6, 6
B. 3, 3
C. 12, 6
D. 32, 24

Your Answer: C Correct Answer: B Incorrect Time Taken: 00m 00s Skipped

Copyright © Stuff