PL-2022-Lab-19-MCQ

201501050@rajalakshmi.edu.in Switch account

Draft saved

Your email will be recorded when you submit this form

* Required

Find the output of the C code given below: *

```
#include <stdio.h>

int main()

{

    int ary[4] = { 1, 2, 3, 4 };

    int *p;

    p = ary + 3;

    *p = 5;

    printf("%d", ary[3]);

    return 0;

}
```

- ○ 2
- ○ 4
- ○ 7
- ◉ 5

What will be the output of following program? *

```
#include <stdio.h>

int main()

{

    int *ptr;

    *ptr = 5;

    printf("%d", *ptr);

    return 0;

}
```

- ○ compilation error
- ◉ Runtime error
- ○ 5
- ○ linker error

Find the output of the following C program. *

```
#include <stdio.h>

fun(char *k)

{

    printf("%s", k);

}

int main()

{

    char s[] = "hello";

    fun(s);

    return 0;

}
```

- (●) hello
- ( ) h
- ( ) compilation error
- ( ) No output

What will be the output of the C program? *

```
#include <stdio.h>

#include <stdlib.h>

int main()

{

    int *ptr;

    *ptr = 26;

    *ptr = 62;

    printf("%d\n", *ptr);

    return 0;

}
```

- ○ 26
- ○ 62
- ◉ Compiler error
- ○ Garbage value

Find the output of the following program: *

```
#include <stdio.h>

int main()

{

    int *ptr, a = 5;

    ptr = &a;

    *ptr += 2;

    printf("%d, %d ", *ptr, a);

    return 0;

}
```

- (●) 7, 7
- ( ) 5, 7
- ( ) 7, 5
- ( ) 5, 5

What is the output of this C code? *

```
#include <stdio.h>

int main()

{

    int i = 97, *p = &i;

    foo(&i);

    printf("%d", *p);

    return 0;

}


void foo(int *p)

{

    int j = 2;

    p = &j;

    printf("%d ", *p);

}
```

- (●) 2 97
- ( ) 2 2
- ( ) Compile time error
- ( ) Segmentation fault/code crash

What is the missing statement in the following which copies string x into string y? *

```
void strcpy(char *x, char *y)
{
    while(*y != '\0')
        . . . /* missing statement */
    *x = '\0';
}
```

What will be the result of execution?

- ○ x=y
- ◉ *x++ = *y++
- ○ (*x)++ = (*y)++
- ○ None of the mentioned

What is the output? *

```
#include <stdio.h>

int main()

{

    char *s = "REC";

    char *p = s;

    printf("%c%c", *(p + 1), s[1]);

    return 0;

}
```

- ○ EC
- ◉ EE
- ○ RE
- ○ RR

What is the output of this C code? *

```
#include <stdio.h>

int main()

{

    int a[3] = {1, 2, 3};

    int *p = a;

    int **r = &p;

    printf("%d", **r);

    return 0;

}
```

- (●) 1
- ( ) Compile time error
- ( ) Address of a
- ( ) Junk value

What will be output when you will execute following c code? *

```
#include<stdio.h>

int main()

{

    short num[3][2] = { 2, 7, 10, 12, 15, 18 };

    printf("%d %d", *(num + 1)[1], **(num + 2));

    return 0;

}
```

- ○ 12 18
- ○ 18 18
- ◉ 15 15
- ○ 12 15

Find the output of the C code given below: *

```
#include <stdio.h>

int main()

{

    int ary[4] = { 1, 2, 3, 4 };

    int *p;

    p = ary + 3;

    *p = 5;

    printf("%d\n", ary[3]);

    return 0;

}
```

- ○ 2
- ○ 4
- ○ 7
- ◉ 5

Find the output of the C program given below. *

```
#include <stdio.h>

int main()

{

    char s1[7] = "1234", *p;

    p = s1 + 2;

    *p = '0';

    printf("%s", s1);

    return 0;

}
```

- (●) 1204
- ( ) 1034
- ( ) 12340
- ( ) 1234

What is the output of this code? *

```
#include <stdio.h>

int *f();

int main()

{

    int *p = f();

    printf("%d", *p);

    return 0;

}

int *f()

{

    int j = 10;

    return &j;

}
```

- ○ 10
- ○ Compile time
- ○ 0
- ◉ Undefined behaviour

What is the output of the following C program? *

```
#include <stdio.h>

int main()

{

    int *p, a = 10;

    p = &10;

    printf("%d", *p);

    return 0;

}
```

- ○ 10
- ○ a
- ○ address of a
- ◉ compilation error

What will be the output of the following? *

```
#include <stdio.h>

int main()

{

    char *ptr;

    char myString[] = "abcdefg";

    ptr = myString;

    ptr += 5;

    printf("%s", ptr);

    return 0;

}
```

- (●) fg
- ( ) efg
- ( ) defg
- ( ) cdefg

What will be the output of following program? *

```
#include <stdio.h>

int main()

{

    int i = 25;

    int *j;

    int **k;

    j = &i;

    k = &j;

    printf("%u %u %u", k, *k, **k);

    return 0;

}
```

- (●) address address value
- ( ) address value value
- ( ) address address address
- ( ) compilation error

What is the output of the following C program? *

```
#include <stdio.h>
struct p
{
    int x;
    char y;
};
int main()
{
    struct p p1[] = { 1, 90, 62, 33, 3, 34 };
    struct p *ptr1 = p1;
    int x = (sizeof(p1) / 3);
    if (x == sizeof(int) + sizeof(char))
        printf("True");
    else
        printf("False");
    return 0;
}
```

- ○ True
- ◉ False
- ○ No output
- ○ Compilation error

What is the output of this C code? *

```
#include <stdio.h>

int main()

{

    int a[3] = {1, 2, 3};

    int *p = a;

    int **r = &p;

    printf("%p %p", *r, a);

    return 0;

}
```

- ○ Different address is printed
- ○ 1 2
- ◉ Same address is printed.
- ○ 1 1

Output of the following program will be? (Assume that integer allocates 4 bytes of memory) *

```
#include <stdio.h>

int main()

{

    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

    int *p = a + 1;

    int *q = a + 6;

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

    return 0;

}
```

- ○ 9
- ◉ 5
- ○ 2
- ○ None of the mentioned

What is the output? *

```
#include <stdio.h>

int main()

{

    char *s = "hello";

    char *p = s;

    printf("%c %c", *(p + 1), s[1]);

    return 0;

}
```

- ○ h e
- ○ e l
- ○ h h
- ◉ e e

What will be the output when you execute the following C code? *

```
#include <stdio.h>

int main()

{

    short num[3][2]={2, 7, 10, 12, 15, 18};

    printf("%d %d",*(num + 1)[1],**(num + 2));

    return 0;

}
```

- ○ 12 18
- ○ 18 18
- ◉ 15 15
- ○ 12 15

What will be output when you will execute following c code? *

```
#include <stdio.h>

int main()

{

    short arr[3][3] = { 2, 7, 10, 12, 15, 18, 25, 32, 35 };

    printf("%d, %d", *(arr + 1)[1], **(arr + 1));

    return 0;

}
```

- ○ 15, 7
- ○ 25, 15
- ◉ 25, 12
- ○ 12, 7

What is the output of the following code snippet? *

```
#include <stdio.h>

int main()

{

    int arr[5] = { 1, 2, 3, 4, 5 };

    int *ptr = arr;

    printf("%d", *ptr);

    return 0;

}
```

- ○ 2
- ○ 5
- ◉ 1
- ○ 4

Find the output of the C code given below: *

```
#include <stdio.h>

int main()

{

    char *s = "hello";

    char *p = s;

    printf("%p\t%p", p, s);

    return 0;

}
```

- ○ Different address is printed
- ◉ Same address is printed
- ○ Run time error
- ○ Nothing

What if the output of the following C program? *

```
#include <stdio.h>

void m(int *p, int *q)

{

    int temp = *p;

    *p = *q;

    *q = temp;

}


int main()

{

    int a = 6, b = 5;

    m(&a, &b);

    printf("%d %d", a, b);

    return 0;

}
```

- (●) 5 6
- ( ) 6 5
- ( ) 5 5
- ( ) 6 6

What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires 4 bytes of memory. *

```
#include <stdio.h>

int main()

{

    unsigned int x[4][3] = {{1,2,3}, {4,5,6}, {7,8,9}, {10,11,12}};

    printf("%u, %u, %u", x + 3, *(x + 3), *(x + 2) + 3);

    return 0;

}
```

- (●) 2036 2036 2036
- ( ) 2012 4 2204
- ( ) 2036 10 10
- ( ) 2012 4 6

What is the output of this C code? *

```
#include <stdio.h>

int main()

{

    int *ptr, a = 10;

    ptr = &a;

    *ptr += 1;

    printf("%d %d", *ptr, a);

    return 0;

}
```

- ○ 10 10
- ○ 10 11
- ○ 11 10
- ◉ 11 11

Find the output of the C program given below: *

```
#include <stdio.h>

int main()

{

    char s1[7] = "123456", *p1, *p2, *p3;

    p1 = s1 + 2;

    p2 = s1 + 3;

    p3 = s1 + 1;

    *p1 = '08';

    *p3 = '8';

    *p2 = '2';

    printf("%s", s1);

    return 0;

}
```

- (●) 188256
- ( ) 123488
- ( ) 188456
- ( ) 123828

What is the output of the following program? *

```
#include <stdio.h>

int main()

{

    int *ptr;

    int x;

    ptr = &x;

    *ptr = 5;

    printf("x=%d\n", x);

    printf("*ptr=%d", *ptr);

    return 0;

}
```

- ○ x=garbage value *ptr=5
- ◉ x=5 *ptr=5
- ○ x=5 *ptr=garbage value
- ○ None of the mentioned

What will be the output of the program assuming that the array begins at the location 1002 and size of an integer is 4 bytes? *

```
#include <stdio.h>

int main()

{

    int a[3][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

    printf("%u, %u, %u", a[0] + 1, *(a[0] + 1), *(*(a + 0) + 1));

    return 0;

}
```

- ○ 448, 4, 4
- ◉ 520, 2, 2
- ○ 1006, 2, 2
- ○ Error

Submit Clear form