You're signed in

You can only respond to this form once. Continue?

SWITCH ACCOUNT

first_Term_EXAM_PART3

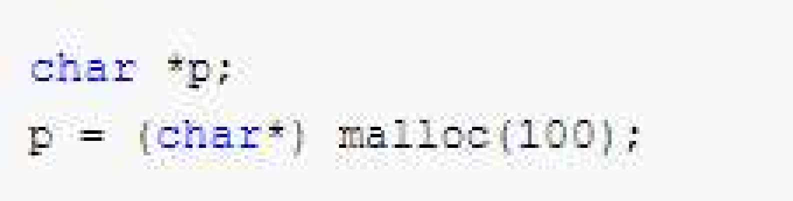
abdallah.shabaan.ghazy@gmail.com Switch account

* Indicates required question

Email *

abdallah.shabaan.ghazy@gmail.com

Q16: Can you combine the following two statements into one? * 1 point

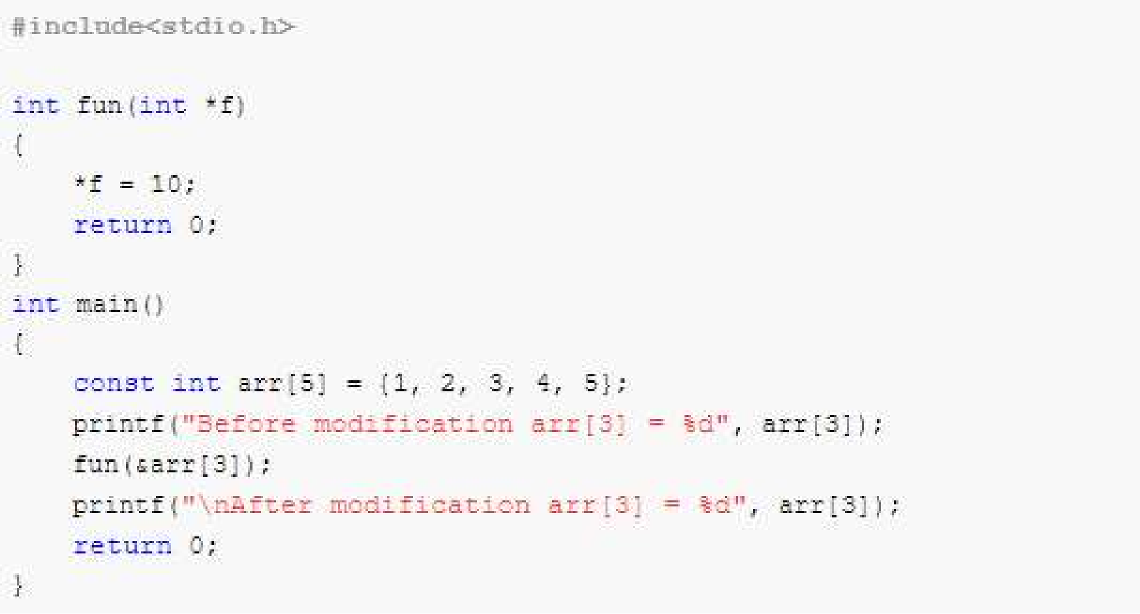


```
char *p;
p = (char*) malloc(100);
```

char *p = (char*) malloc(100);

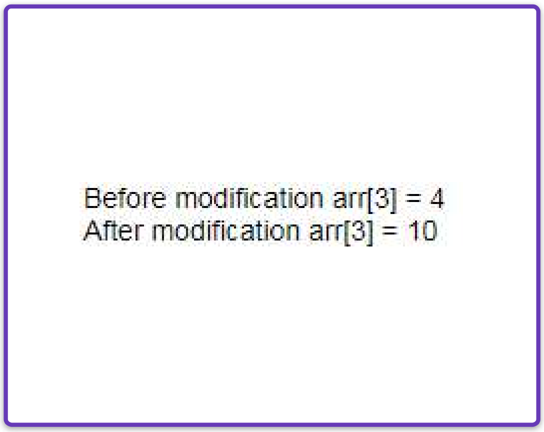
Q25: *

1 point



```
#include<stdio.h>

int fun(int *f)
{
    *f = 10;
    return 0;
}
int main()
{
    const int arr[5] = {1, 2, 3, 4, 5};
    printf("Before modification arr[3] = %d", arr[3]);
    fun(&arr[3]);
    printf("\nAfter modification arr[3] = %d", arr[3]);
    return 0;
}
```



```
Before modification arr[3] = 4
After modification arr[3] = 10
```

● A.

○ B. Error: cannot convert parameter 1 from const int * to int *

○ C. Error: Invalid parameter

○ D. Before modification arr[3] = 4

○ After modification arr[3] = 4

Q31:Which of the following is true about linked list implementation of stack? * 1 point

○ In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.

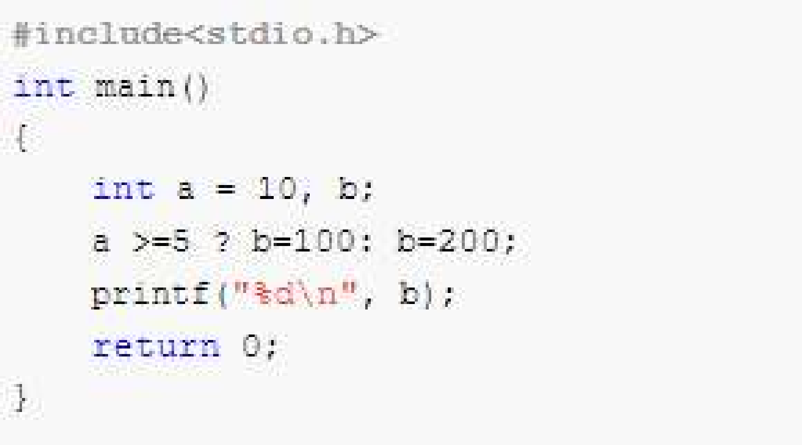
○ In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.

○ Both of the above

● None of the above

Q17: *

1 point



```
#include<stdio.h>
int main()
{
    int a = 10, b;
    a >=5 ? b=100: b=200;
    printf("%d\n", b);
    return 0;
}
```

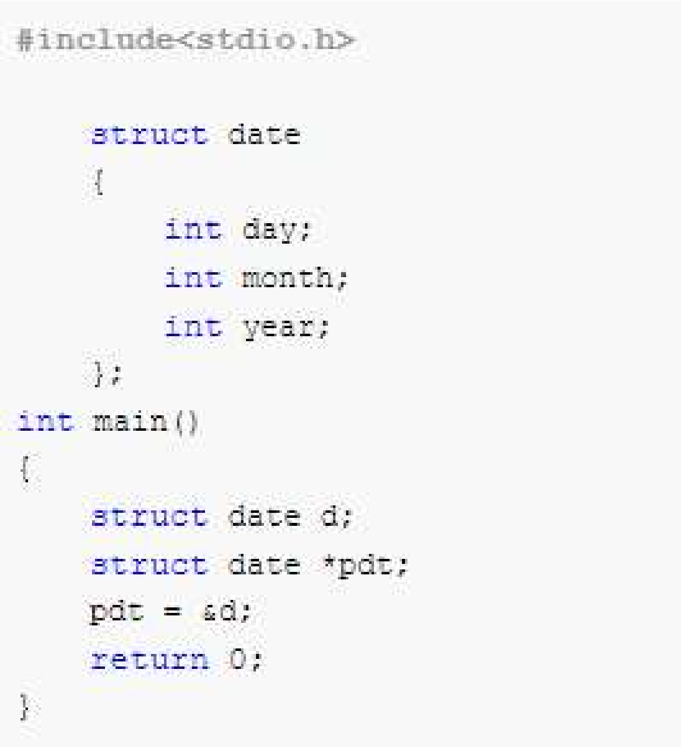
○ A. 100

○ B. 200

● C. Error: L value required for b

○ D. Garbage value

Q22:Which of the following statements correctly assigns 12 to month using pointer variable pdt? * 1 point



```
#include<stdio.h>

    struct date
    {
        int day;
        int month;
        int year;
    };
int main()
{
    struct date d;
    struct date *pdt;
    pdt = &d;
    return 0;
}
```

○ A. pdt.month = 12

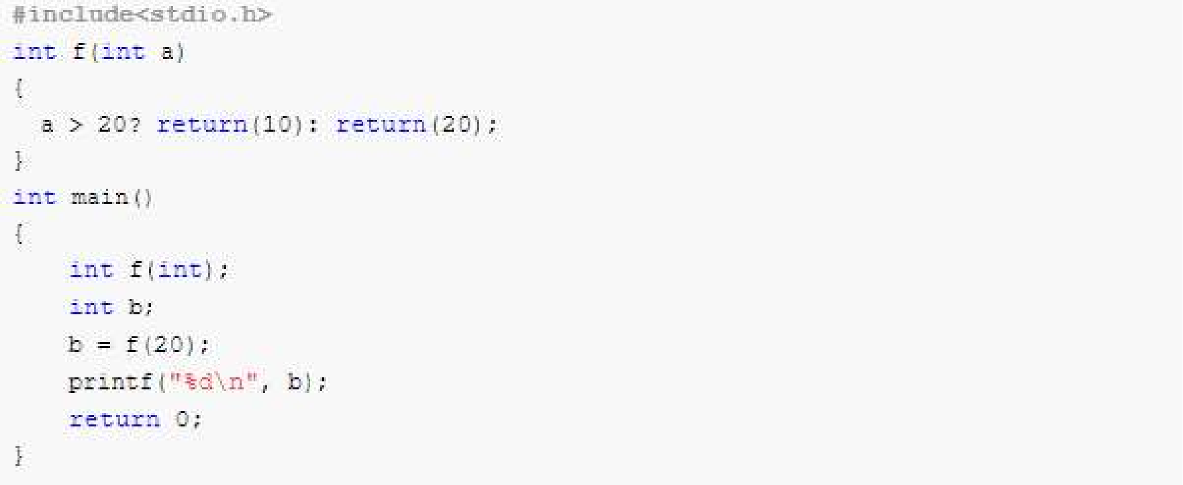
○ B. &pdt.month = 12

○ C. d.month = 12

● D. pdt->month = 12

Q9:Point out the error in the program *

1 point



```
#include<stdio.h>
int f(int a)
{
  a > 20? return(10): return(20);
}
int main()
{
    int f(int);
    int b;
    b = f(20);
    printf("%d\n", b);
    return 0;
}
```

○ A. Error: Prototype declaration

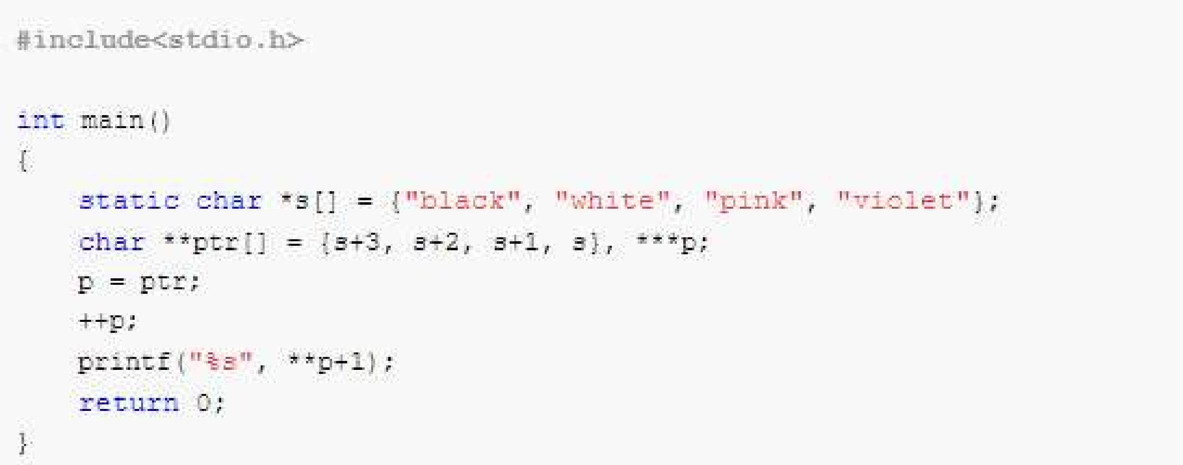
○ B. No error

● C. Error: return statement cannot be used with conditional operators

○ D. None of above

Q18: *

1 point



```
#include<stdio.h>

int main()
{
    static char *s[] = {"black", "white", "pink", "violet"};
    char **ptr[] = {s+3, s+2, s+1, s}, ***p;
    p = ptr;
    ++p;
    printf("%s", **p+1);
    return 0;
}
```

● ink

○ pink

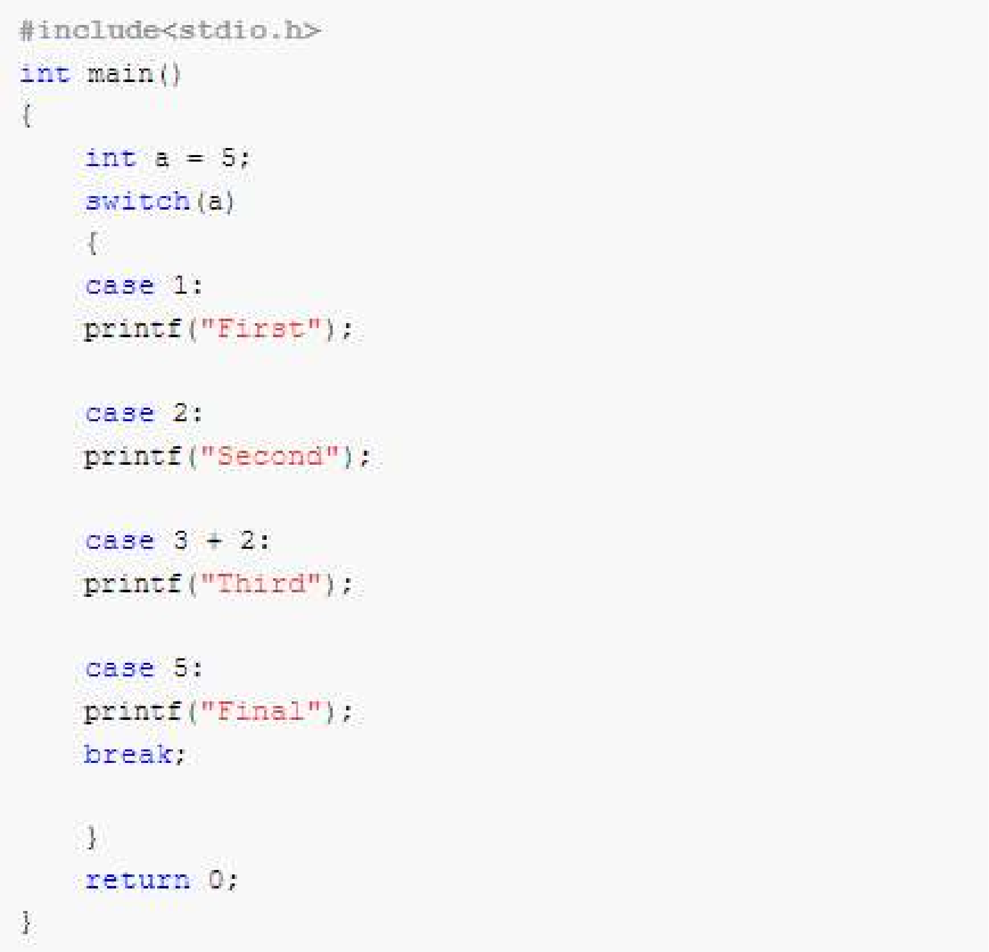
○ black

○ violet

○ hite

○ white

Q20: * 1 point



```
#include<stdio.h>
int main()
{
    int a = 5;
    switch(a)
    {
    case 1:
    printf("First");

    case 2:
    printf("Second");

    case 3 + 2:
    printf("Third");

    case 5:
    printf("Final");
    break;

    }
    return 0;
}
```

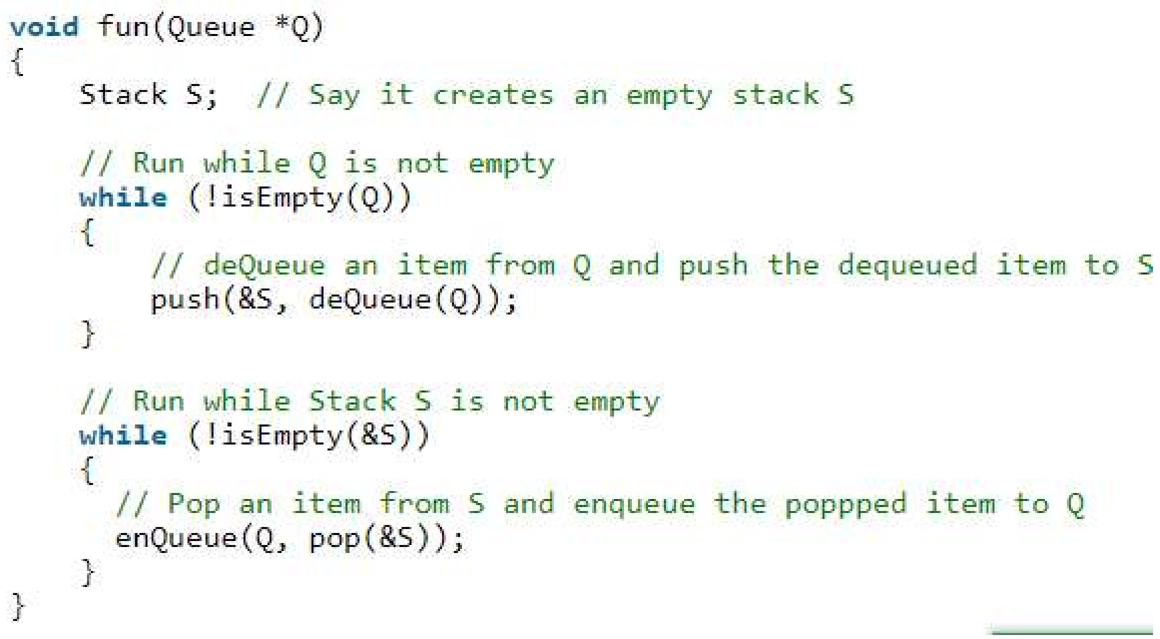
○ A. There is no break statement in each case.

○ B. Expression as in case 3 + 2 is not allowed.

● C. Duplicate case case 5:

○ D. No error will be reported.

Q32:What does the below function do in general? * 1 point



```
void fun(Queue *Q)
{
    Stack S;  // Say it creates an empty stack S

    // Run while Q is not empty
    while (!isEmpty(Q))
    {
        // deQueue an item from Q and push the dequeued item to S
        push(&S, deQueue(Q));
    }

    // Run while Stack S is not empty
    while (!isEmpty(&S))
    {
      // Pop an item from S and enqueue the poppped item to Q
      enQueue(Q, pop(&S));
    }
}
```

○ Removes the last from Q

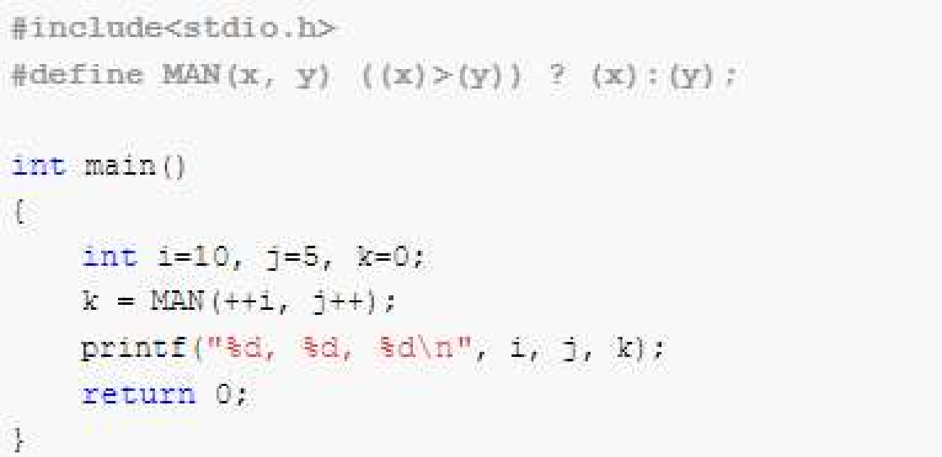
● Keeps the Q same as it was before the call

○ Makes Q empty

○ Reverses the Q

Q27: *

1 point



```
#include<stdio.h>
#define MAN(x, y) ((x)>(y)) ? (x):(y);

int main()
{
    int i=10, j=5, k=0;
    k = MAN(++i, j++);
    printf("%d, %d, %d\n", i, j, k);
    return 0;
}
```

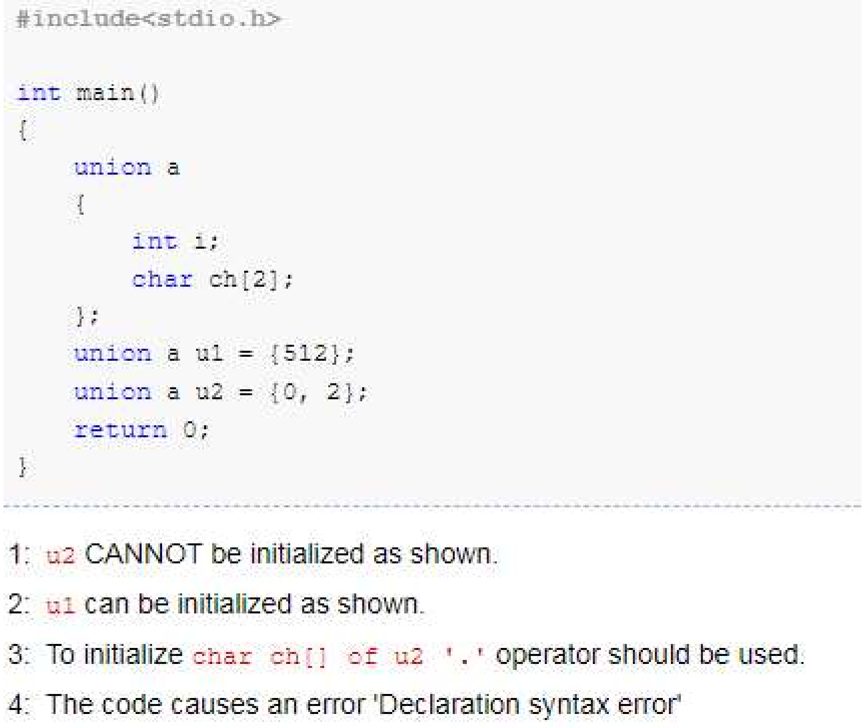
● A. 12, 6, 12

○ B. 11, 5, 11

○ C. 11, 5, Garbage

○ D. 12, 6, Garbage

Q21:Which of the following statements correct about the below program? * 1 point



```
#include<stdio.h>

int main()
{
    union a
    {
        int i;
        char ch[2];
    };
    union a u1 = {512};
    union a u2 = {0, 2};
    return 0;
}
```

1: u2 CANNOT be initialized as shown.

2: u1 can be initialized as shown.

3: To initialize char ch[] of u2 '.' operator should be used.

4: The code causes an error 'Declaration syntax error'

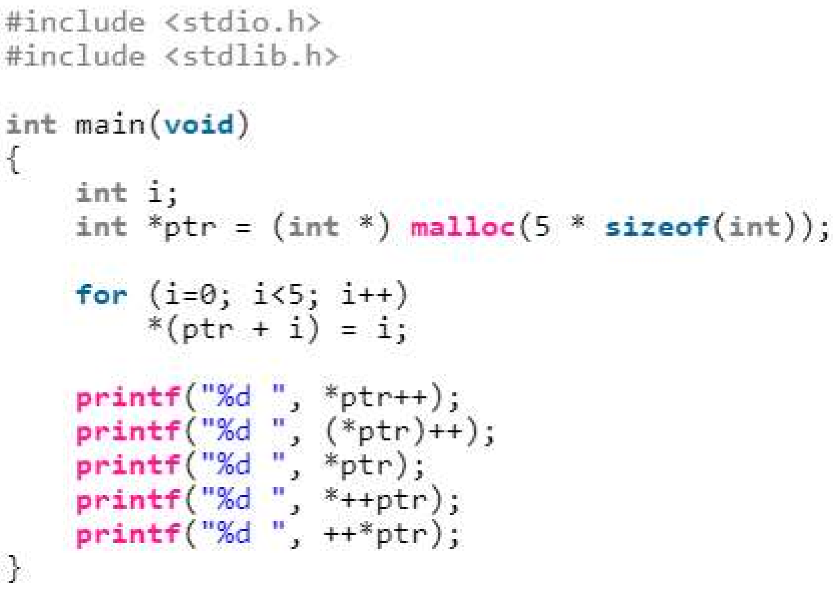
○ A. 1, 2

○ B. 2, 3

● C. 1, 2, 3

○ D. 1, 3, 4

Q34: * 1 point



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

int main(void)
{
    int i;
    int *ptr = (int *) malloc(5 * sizeof(int));

    for (i=0; i<5; i++)
        *(ptr + i) = i;

    printf("%d ", *ptr++);
    printf("%d ", (*ptr)++);
    printf("%d ", *ptr);
    printf("%d ", *++ptr);
    printf("%d ", ++*ptr);
}
```

○ Compiler Error

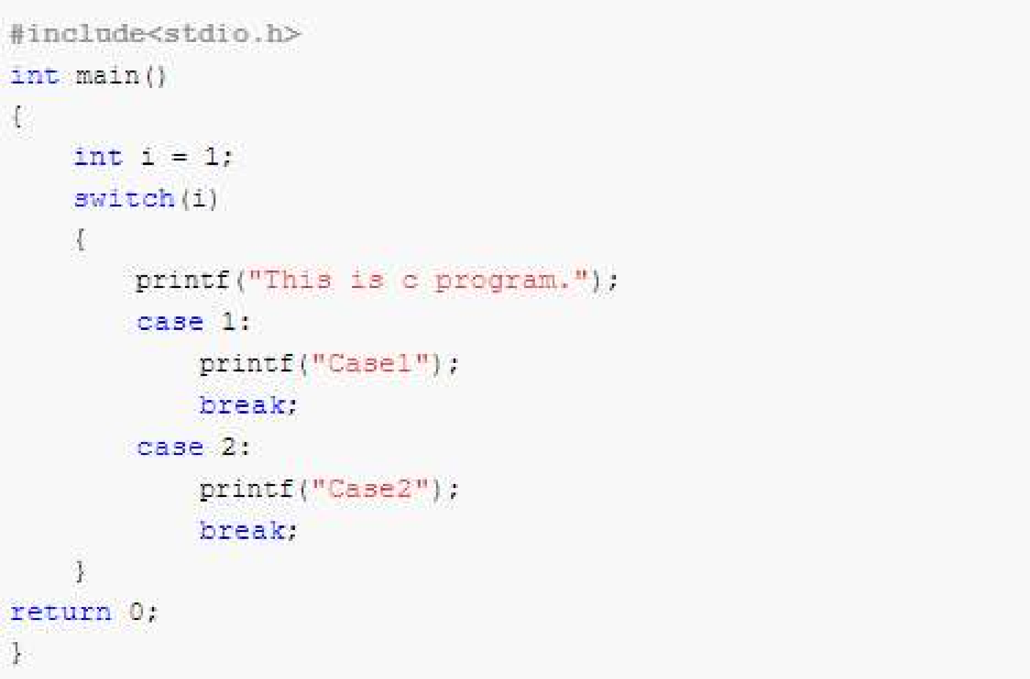
● 0 1 2 2 3

○ 0 1 2 3 4

○ 1 2 3 4 5

Q26: *

1 point



```
#include<stdio.h>
int main()
{
    int i = 1;
    switch(i)
    {
        printf("This is c program.");
        case 1:
            printf("Case1");
            break;
        case 2:
            printf("Case2");
            break;
    }
return 0;
}
```

○ A. Error: No default specified

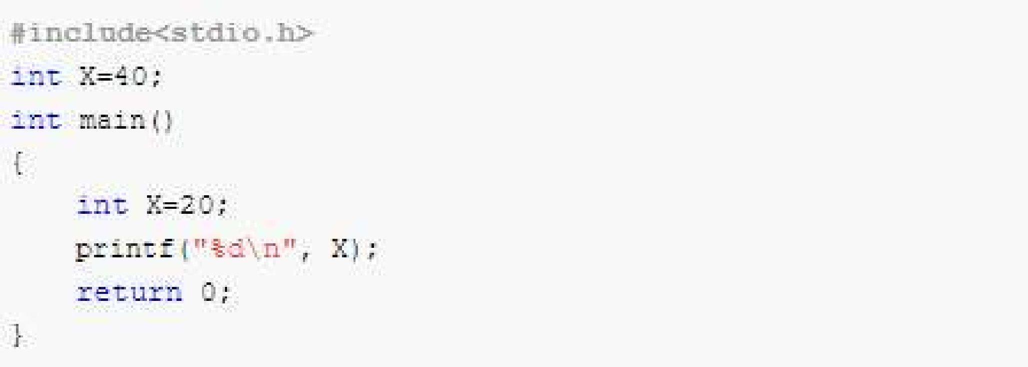
○ B. Error: Invalid printf statement after switch statement

● C. No Error and prints "Case1"

○ D. None of above

Q5:What is the output of the program *

1 point



```
#include<stdio.h>
int X=40;
int main()
{
    int X=20;
    printf("%d\n", X);
    return 0;
}
```

● A. 20

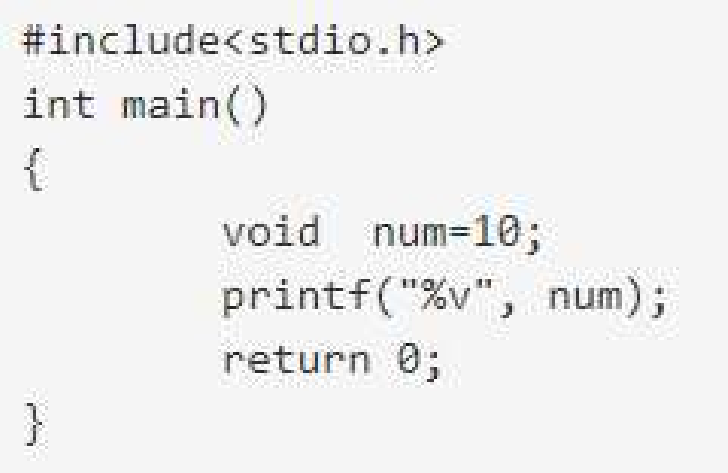
○ B. 40

○ C. compilation error

○ D. No Output

Q3: What will be the output of the C program? *

1 point



```
#include<stdio.h>
int main()
{
        void  num=10;
        printf("%v", num);
        return 0;
}
```

● Compilation error

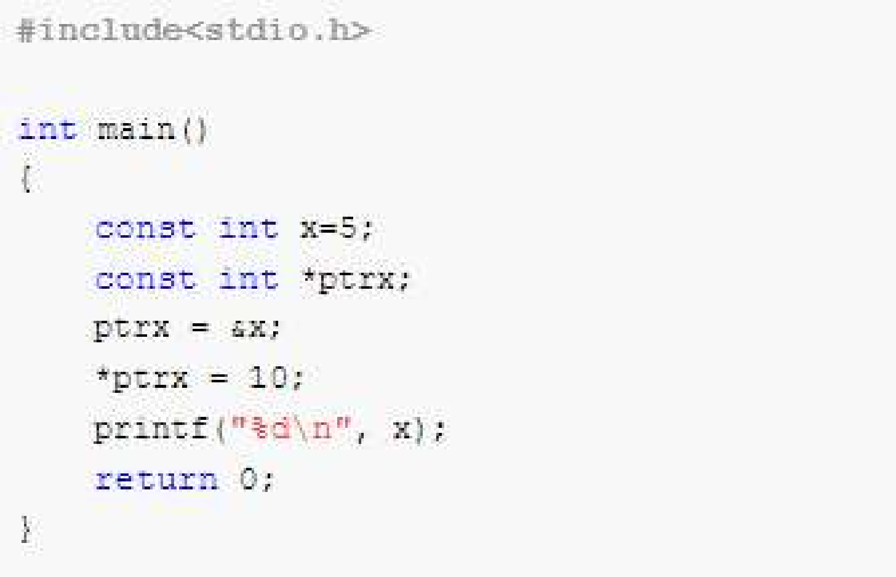
○ 10

○ Garbage value

○ 0

Q19: *

1 point



```
#include<stdio.h>

int main()
{
    const int x=5;
    const int *ptrx;
    ptrx = &x;
    *ptrx = 10;
    printf("%d\n", x);
    return 0;
}
```

○ A. 5

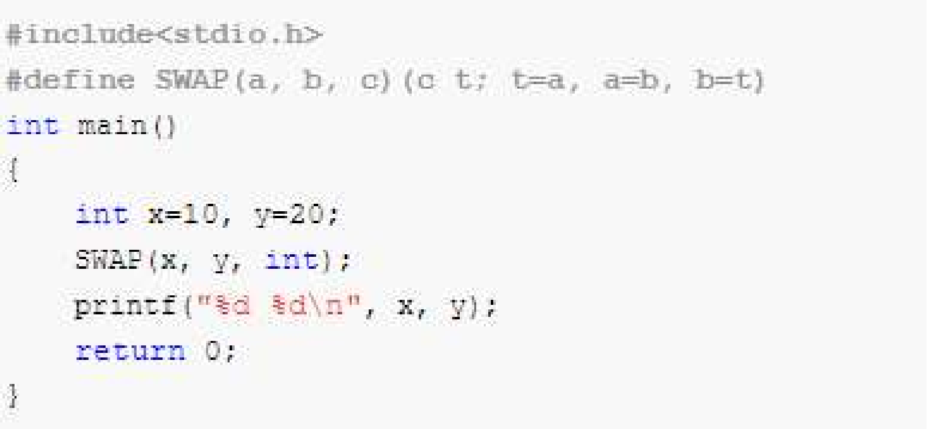
● B. 10

○ C. Error

○ D. Garbage value

Q11: *

1 point



```
#include<stdio.h>
#define SWAP(a, b, c)(c t; t=a, a=b, b=t)
int main()
{
    int x=10, y=20;
    SWAP(x, y, int);
    printf("%d %d\n", x, y);
    return 0;
}
```

○ 20 10

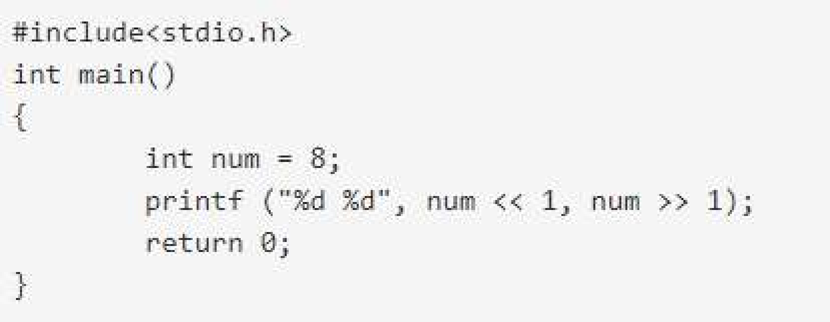
○ compile but no output

● not compile

○ compile with warning

Q2: What will be the output of the C program? *

1 point



```
#include<stdio.h>
int main()
{
        int num = 8;
        printf ("%d %d", num << 1, num >> 1);
        return 0;
}
```

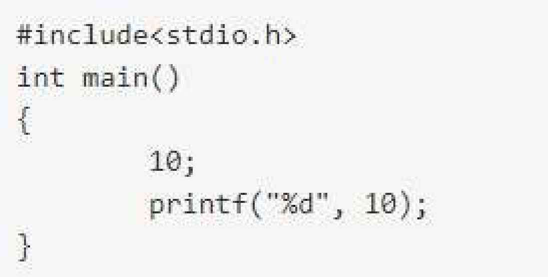
○ 9 7

○ 7 9

● 16 4

○ 4 16

Q6:What is the output of the program? * 1 point



```
#include<stdio.h>
int main()
{
        10;
        printf("%d", 10);
}
```

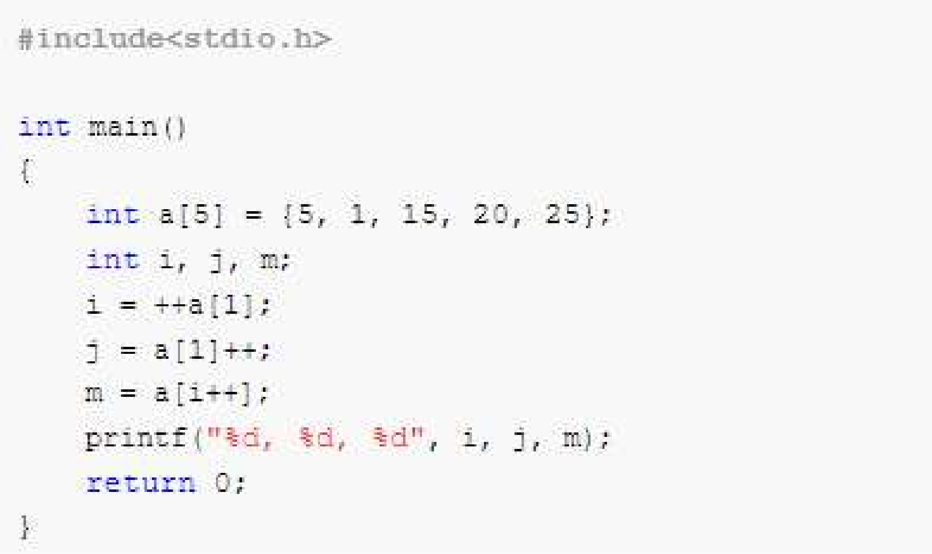
○ Compilation Error

● 10

○ Runtime error

○ No output

Q10: * 1 point



```
#include<stdio.h>

int main()
{
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    printf("%d, %d, %d", i, j, m);
    return 0;
}
```

○ A. 2, 1, 15

○ B. 1, 2, 5

● C. 3, 2, 15

○ D. 2, 3, 20

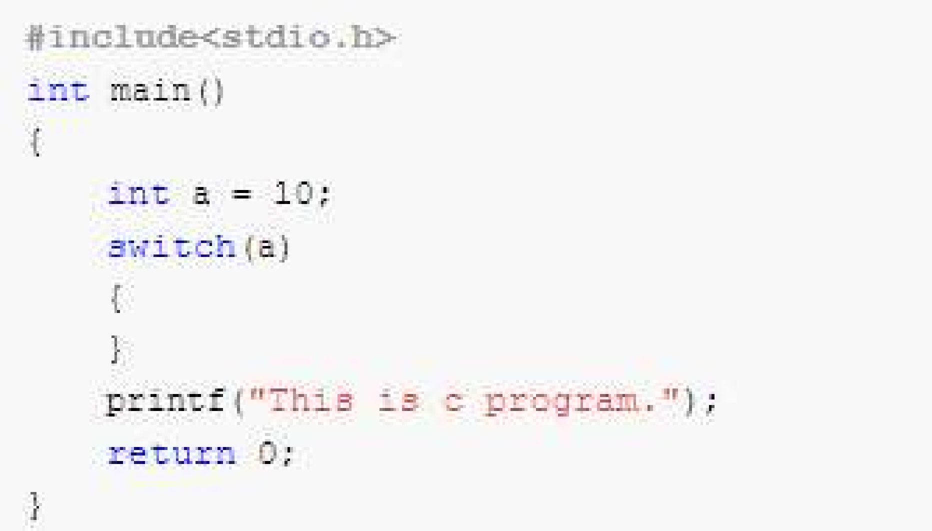
○ E.15, 20, garbage value

○ F. 2, 2, 1

○ G. 2, 2, 15

○ Other:

Q13: * 1 point



```
#include<stdio.h>
int main()
{
    int a = 10;
    switch(a)
    {
    }
    printf("This is c program.");
    return 0;
}
```

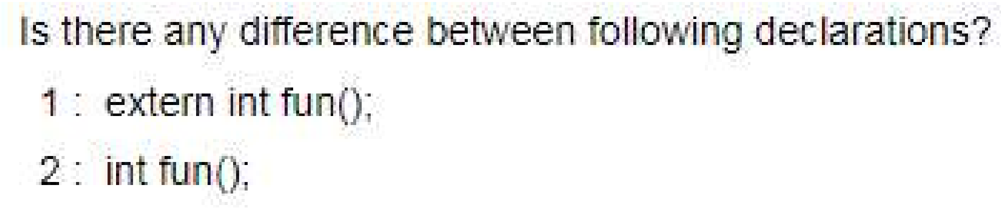
○ A. Error: No case statement specified

○ B. Error: No default specified

● C. No Error

○ D. Error: infinite loop occurs

Q4: * 1 point



Is there any difference between following declarations?

1 : extern int fun();

2 : int fun();

○ A. Both are identical

● B. No difference, except extern int fun(); is probably in another file

○ C. int fun(); is overrided with extern int fun();

○ D. None of these

Q15: What is (void*)0? *

1 point

● A. Representation of NULL pointer

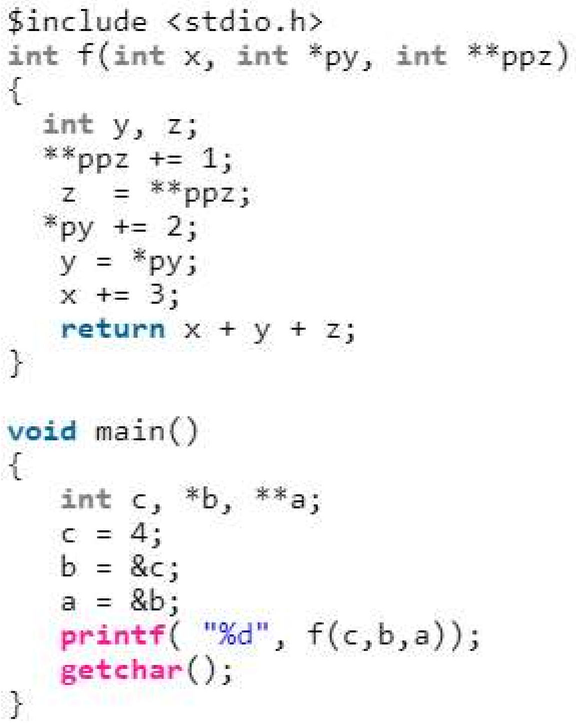
○ B. Representation of void pointer

○ C. Error

○ D. None of above

Q35:What is printed by the following C program? *

1 point



```
$include <stdio.h>
int f(int x, int *py, int **ppz)
{
  int y, z;
  **ppz += 1;
   z  = **ppz;
  *py += 2;
   y = *py;
   x += 3;
   return x + y + z;
}

void main()
{
   int c, *b, **a;
   c = 4;
   b = &c;
   a = &b;
   printf( "%d", f(c,b,a));
   getchar();
}
```

○ 18

● 19

○ 21

○ 22

Q12: Which of the following cannot be checked in a switch-case statement? * 1 point

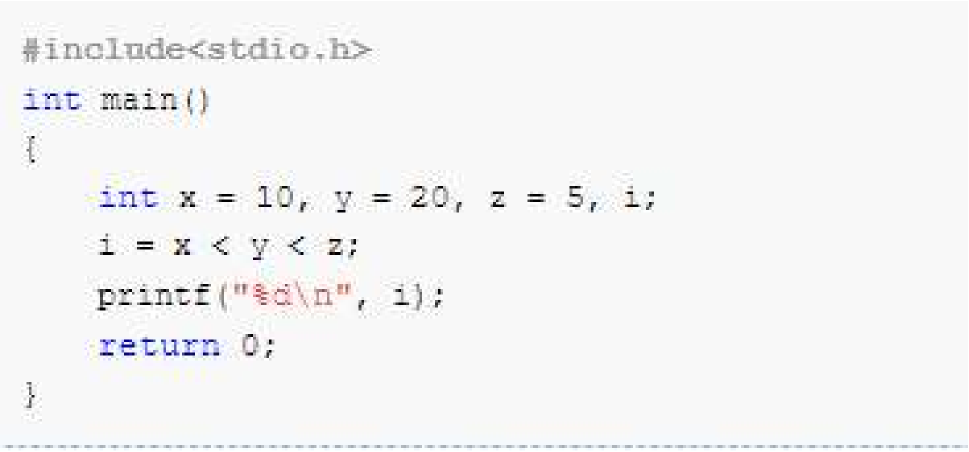
○ A. Character

○ B. Integer

● C. Float

○ D. enum

Q30:What is the output of the program? * 1 point



```
#include<stdio.h>
int main()
{
    int x = 10, y = 20, z = 5, i;
    i = x < y < z;
    printf("%d\n", i);
    return 0;
}
```

○ A. 0

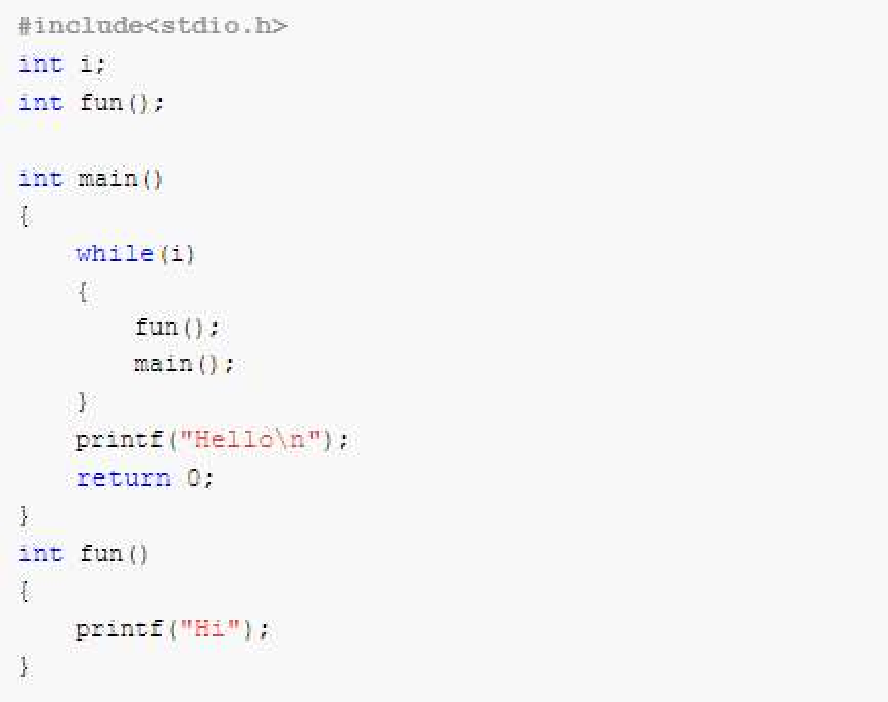
● B. 1

○ C. Error

○ D. None of these

Q29: *

1 point



```
#include<stdio.h>
int i;
int fun();

int main()
{
    while(i)
    {
        fun();
        main();
    }
    printf("Hello\n");
    return 0;
}
int fun()
{
    printf("Hi");
}
```

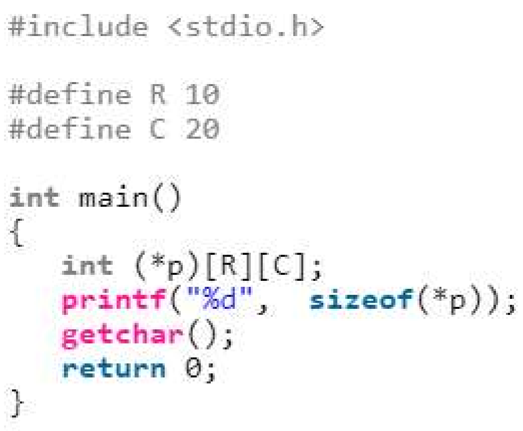
● A. Hello

○ B. Hi Hello

○ C. No output

○ D. Infinite loop

Q33:Assume sizeof an integer and a pointer is 4 byte. Output? * 1 point



```
#include <stdio.h>

#define R 10
#define C 20

int main()
{
   int (*p)[R][C];
   printf("%d",  sizeof(*p));
   getchar();
   return 0;
}
```

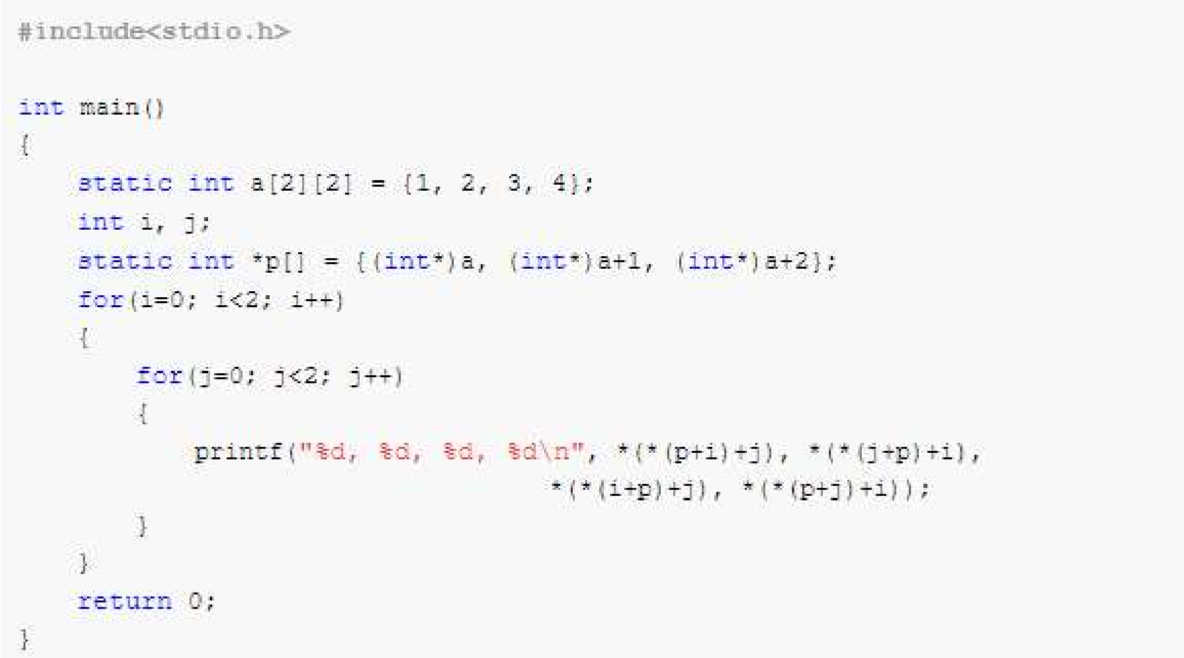
○ 200

● 4

○ 800

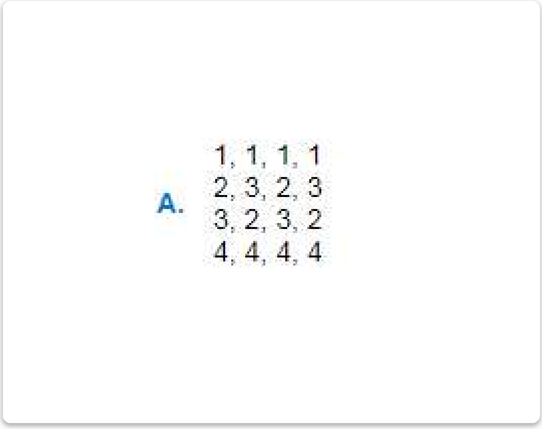
○ 80

Q28: * 1 point

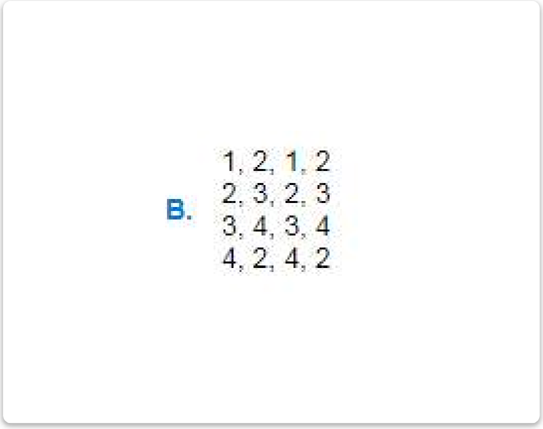


```
#include<stdio.h>

int main()
{
    static int a[2][2] = {1, 2, 3, 4};
    int i, j;
    static int *p[] = {(int*)a, (int*)a+1, (int*)a+2};
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("%d, %d, %d, %d\n", *(*(p+i)+j), *(*(j+p)+i),
                                    *(*(i+p)+j), *(*(p+j)+i));
        }
    }
    return 0;
}
```



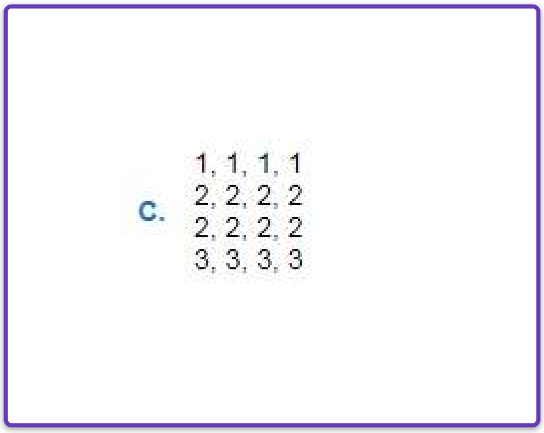
A.
```
1, 1, 1, 1
2, 3, 2, 3
3, 2, 3, 2
4, 4, 4, 4
```



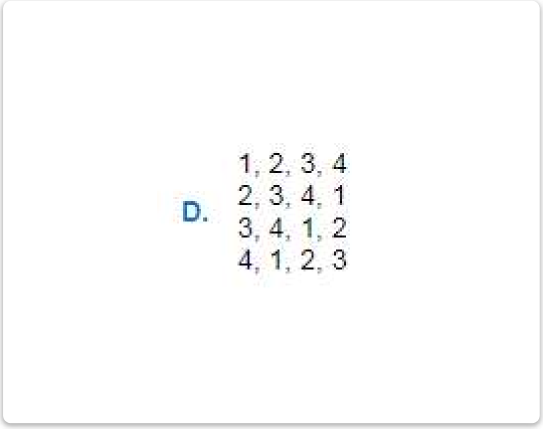
B.
```
1, 2, 1, 2
2, 3, 2, 3
3, 4, 3, 4
4, 2, 4, 2
```

○ A

○ B



C.
```
1, 1, 1, 1
2, 2, 2, 2
2, 2, 2, 2
3, 3, 3, 3
```

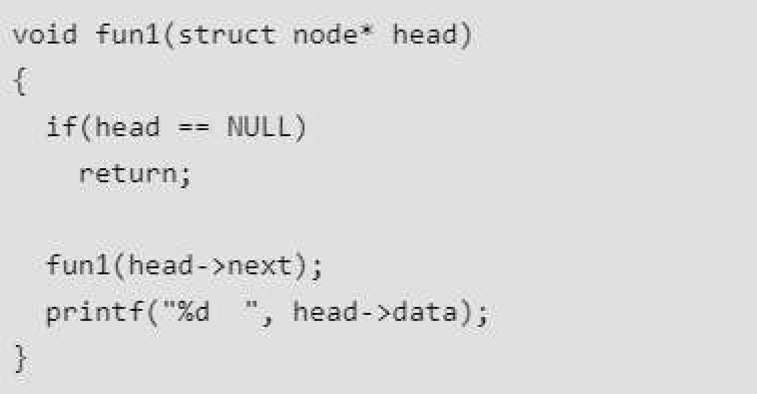


D.
```
1, 2, 3, 4
2, 3, 4, 1
3, 4, 1, 2
4, 1, 2, 3
```

● C

○ D

Q24:What does the following function do for a given Linked List with first node as head? * 1 point



```
void fun1(struct node* head)
{
  if(head == NULL)
    return;

  fun1(head->next);
  printf("%d  ", head->data);
}
```

● Prints all nodes of linked lists

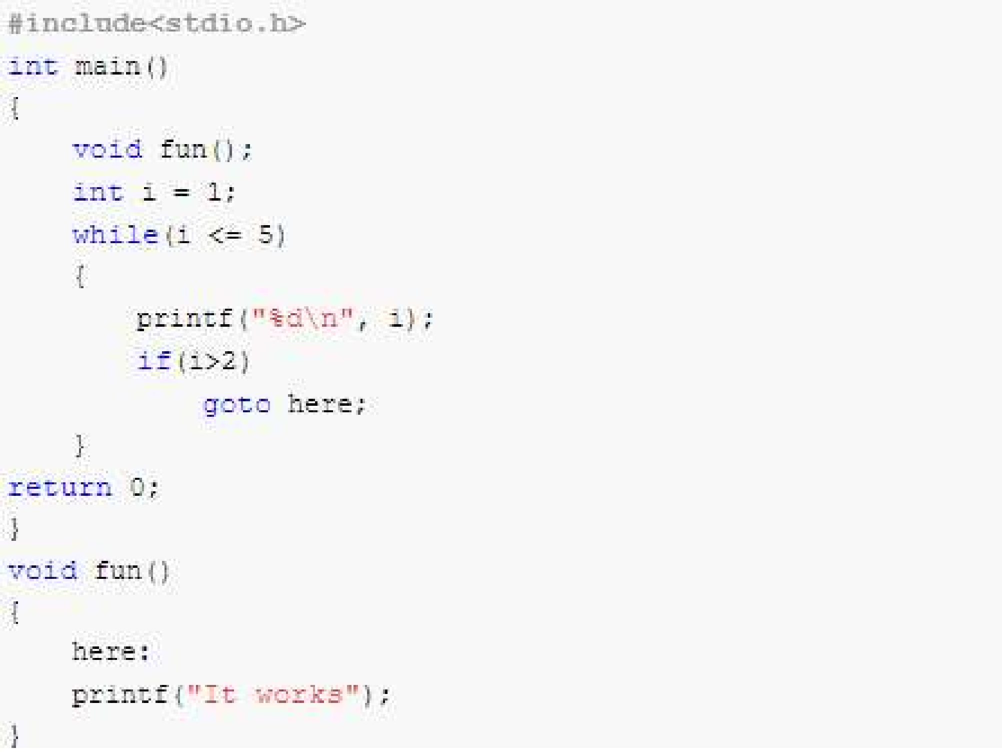
○ Prints all nodes of linked list in reverse order

○ Prints alternate nodes of Linked List

○ Prints alternate nodes in reverse order

Q14: Point out the error, if any in the while loop.

1 point



```
#include<stdio.h>
int main()
{
    void fun();
    int i = 1;
    while(i <= 5)
    {
        printf("%d\n", i);
        if(i>2)
            goto here;
    }
return 0;
}
void fun()
{
    here:
    printf("It works");
}
```

○ A. No Error: prints "It works"

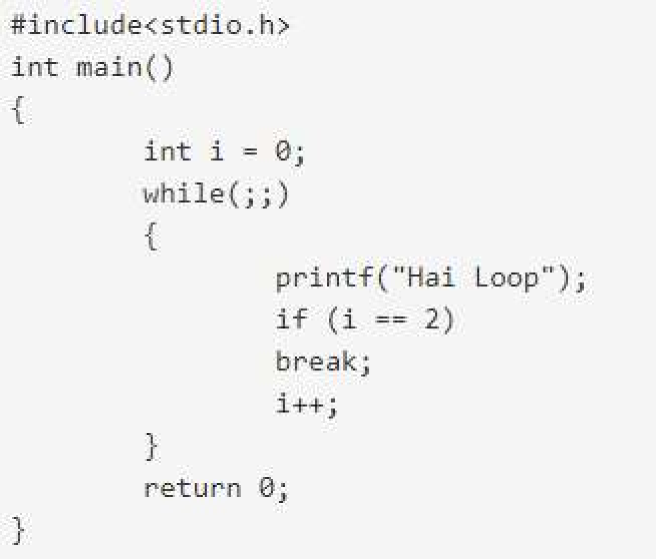
○ B. Error: fun() cannot be accessed

● C. Error: goto cannot takeover control to other function

○ D. No error

Clear selection

Q7:What is the output of the program? * 1 point



```
#include<stdio.h>
int main()
{
        int i = 0;
        while(;;)
        {
                printf("Hai Loop");
                if (i == 2)
                break;
                i++;
        }
        return 0;
}
```

○ A. Hai Loop

● B. Compilation Error

○ C. Hai Loop Hai Loop

○ D. Hai Loop Hai Loop Hai Loop

Q1: What are the types of linkages? * 1 point

● A. Internal and External

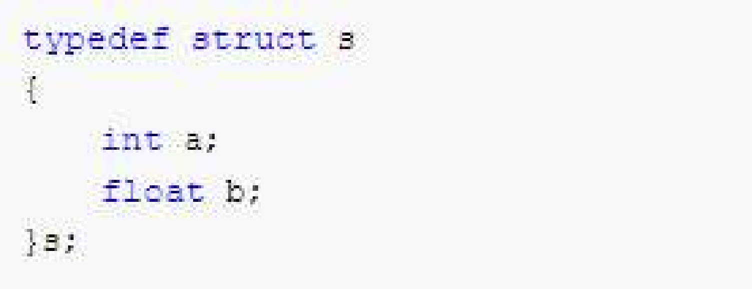
○ B. External, Internal and None

○ C. External and None

○ D. Internal

Q23:Will the following declaration work? *

1 point



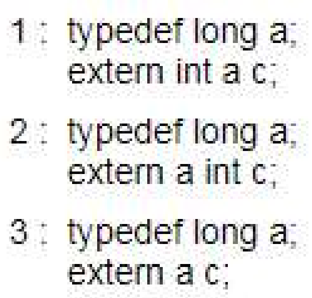
```
typedef struct s
{
    int a;
    float b;
}s;
```

● A. Yes

○ B. No

Q8: Which of the declaration is correct? *

1 point



```
1 : typedef long a;
    extern int a c;
2 : typedef long a;
    extern a int c;
3 : typedef long a;
    extern a c;
```

○ A. 1 correct

● B. 2 correct

○ C. 3 correct

○ D. 1, 2, 3 are correct

Submit

Page 1 of 1

Clear form

Never submit passwords through Google Forms.