1.#include<stdio.h> a() { printf("Function"); } b() { printf("Function in C"); } c() { printf("C function"); } main() { int (\*ptr[3])(); ptr[0] = a; ptr[1] = b; ptr[2] = c; ptr[2](https://github.com/hari-eur/Deepthi---Sudhakar/wiki/Module-2-Exercise-Qns---01-03-2023); return 0; }

A. Function

B. Function in C

**C. C function**

D. None of the above

**Answer: C. C function**

**Explanation: ptr2 is called which is function c, so C function is printed**

2.#include<stdio.h> int function(); main() { int i; i = function(); printf("%d", i); return 0; } function() { int a; a = 250; }

A. Runtime error

B. 0

C. 250

**D. 1**

**Answer: D. 1**

**Explanation: In function( ) no value is returned. If nothing is returned in C by default 1 is returned**

3.#include<stdio.h> int main() { function(); return 0; } void function() { printf("Function in C is awesome"); }

**A. Function in C is awesome**

B. no output

C. Runtime error

D. Compilation error

**Answer: A. Function in C is awesome**

**Explanation: function() is called which prints Function in C is awesome**

4. #include<stdio.h> int main() { int num = \_a\_123(4); printf("%d\n", --num); return 0; } int \_a\_123(int num) { return(num++); }

**A. 3**

B. Compilation error

C. 4

D. 5

**Answer: A. 3**

**Explanation: the function \_a\_123 is called with argument 4. The function returns num++ which returns 4 due to post increment. Then –num is printed which prints 3**

5. #include<stdio.h> int main() { char c = ' ', x; getc(c); if((c >= 'a') && (c <= 'z')) x = convert(c); printf("%c", x); return 0; }

A. Runtime Error

B. Any symbols or special characters

**C. Compilation Error**

D. B

**Answer: C. Compilation Error**

**Explanation: convert() is not declared and getc can be used when pointing to a character in a file**

6. #include<stdio.h> void abc(); int \*ptr; int main() { int i, \*p = &i; abc(); return 0; } void abc() { int i = 0; ptr = &i; ptr++; \*ptr = 3; printf("\nFunction in C %d", i); }

**A. Function in C 0**

B. Function in C 1

C. Function in C 2

D. No output

**Answer: A. Function in C 0**

**Explanation: ptr is incremented and then assigned to 3. So it wont affect the value of i and it remains 0**

7. #include<stdio.h> char normal[15] = "Ambulance"; char accident[15]; int main() { swab(normal, accident, strlen(normal +1)); printf ("%s\n", normal); return 0; }

A. Ambulance

B. mAubalcne

**C. mAubalcn**

D. ecnalubmA

**Answer: C. mAubalcn**

**Explanation: swab is a function in c which is capable of swapping two text consecutively in a string.**

8. #include<stdio.h> void ptr(char\*\*); int main() { char \*argv[] = { "abc", "def", "ghi", "jkl", "mno", "pqr" }; ptr(argv); return 0; } void ptr(char \*\*p) { char \*t; t = (p += sizeof(int))[-1]; printf("%s\n", t); }

A. ghi

**B. jkl**

C. mno

D. pqr

**Answer: B. jkl**

**Explanation: p has the address of abc. p+=sizeof(int)[-1]**

**p+4-1= p+3 which is address of jkl**

9. #include<stdio.h> void fun(int\*, int); void (*ptr[1])(int*, int); int main() { int a = 2; int b = 4; ptr[0] = fun; ptr[0](&a, b); printf("%d %d ", a, b); return 0; } void fun(int \*p, int q) { int tmp = \*p; \*p = q; q = tmp; }

A. 2 2

B. 4 2

**C. 4 4**

D. 2 4

**Answer: C. 4 4**

**Explanation: ptr[0](&a, b); a is called by reference and b is called by value. Any change made in reference will be reflected but any change made in value is not reflected so b remains same and a is changed to 4**

10. #include<stdio.h> int main() { int i = 0; while(i+1) while(i<<2) while(i4) { printf("Loop "); if(i == 3) break; } return 0; }

A. Loop Loop Loop

B. Loop Loop

C. Loop

**D. Infinite Loop**

**Answer: D. Infinite Loop**

**Explanation: while condition is true always so infinite loop is printed**

11. #include<stdio.h> int main() { int i = 0; while(++i) { i == --i?i = 0:i = 1; } printf("%d", i); return 0; }

A. 0

B. Infinite Loop

C. 1

**D. Compilation Error**

**Answer: D. Compilation Error**

12. #include<stdio.h> int main() { int i = 1; do { while(i) i--; for(i++;0;i++); break; }while(1); printf("%d", i); return 0; }

**A. 1**

B. 2

C. 0

D. Infinite Loop

**Answer: A. 1**

**Explanation: do {**

**while(i) here i is 1 so enters loop**

**i--; here i becomes 0**

**for(i++;0;i++); in first iteration i is 0 and then post incremented to 1 and then condition fails and comes out of loop**

**break; the next statement is break so it comes out of do while loop**

**printf("%d", i); so i remains 1 while printing**

13. #include<stdio.h> int main(){ int i = -1; do { printf("HiDoWhile "); }while(i++); return 0; }

A. Compilation Error

B. HiDoWhile

C. HiDoWhile HiDoWhile HiDoWhile

**D. HiDoWhile HiDoWhile**

**Answer: D. HiDoWhile HiDoWhile**

**Explanation: in first iteration the print statement is executed without checking any condition. In second iteration while(i++) the value of i remains -1 since its post incremented so print statement is executed for second time. Then i value becomes 0 the condition fails and loop is terminated.**

14. #include<stdio.h> int main(){ while(!printf("Steve is awesome")); return 0; }

**A. Steve is awesome**

B. Infinite Loop

C. Prints Nothing

D. None of the Above

**Answer: A. Steve is awesome**

**Explanation:** while(!printf("Steve is awesome")); returns 0 and loop is terminated. Still printf statement is executed which is true

15. #include<stdio.h> int main() { int i[3] = {1, 4, 0}; while(i[2] == i[3]) { if(i[3]) printf("Loop "); else break; } return 0; }

A. Compilation Error

B. Runtime Error

**C. Prints Nothing**

D. Loop## ### ## #

**Answer: C. Prints Nothing**

**Explanation: i[2] is not equal to i[3] so the loop is terminated and prints nothing**