# CodeWreck2k17

**(Round 0 : Debugging)**

**Instructions:**

**1. You will be given 30 questions with multiple options.**

**2. You have to choose correct option to all questions in 30 minutes of duration.**

**3. You have to write correct option in the table provided in the last page at respective question number.**

**4. No negative marking system for wrong answers.**

**5. Answers should be clearly written. No marks will be provided for any ambiguous answers.**

**6. Use the white papers provided at the end of booklet for rough work. No additional papers are given.**

**Fill the details given below:**

**1) Name:**

**2) Contact No:**

**3) College:**

**4) Batch id:**

**Questions:**

**1)**

**#include <stdio.h>**

**int main()**

**{**

**char \*p = 0;**

**\*p = 'a';**

**printf("value in pointer p is %c\n", \*p);**

**return 0;**

**}**

**Options:**

**a) It will print a**

**b) It will print 0**

**c) Compile time error**

**d) Run time error**

**2)#include <stdio.h>**

**int main()**

**{**

**char \*p = "code wreck";**

**p[0] = 'v';**

**p[1] = 'i';**

**printf("%s", p);**

**return 0;**

**}**

**Options:**

**a) vide wreck**

**b) code wreck**

**c) compile error**

**d) runtime error**

**3) #include <stdio.h>**

**int main()**

**{**

**int n = 0, m = 0;**

**if (n > 0)**

**if (m > 0)**

**printf("True");**

**else**

**printf("False");**

**return 0;**

**}**

**Options:**

**a) True**

**b) False**

**c) No Output will be printed**

**d) Run Time Error**

**4) #include <stdio.h>**

**int main()**

**{**

**int b = 5 - 4 + 2 \* 5 ;**

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

**return 0;**

**}**

**Options:**

**a) 25**

**b) -5**

**c) 11**

**d) None of the mentioned**

**5) What is the sizeof(char) in a 32-bit C compiler?**

**a. 1 bit**

**b. 2 bits**

**c. 2 nibble**

**d. 1 nibble**

**6) void \* malloc ( size\_t n) returns**

**Options:**

**a) Pointer to n bytes of uninitialized storage**

**b) NULL if the request cannot be satisfied**

**c) Nothing**

**d) Both a & b are true**

**7) #include <stdio.h>**

**int main()**

**{**

**char a[10]="CodeWreck";**

**int i = 0;**

**for( ; a[i] ;i++)**

**printf ( "%c" ,i[a]);**

**return 0;**

**}**

**Options:**

**a. CodeWreck**

**b. C**

**c. kcerWedoC**

**d. Compilation error**

**8) #include<stdio.h>**

**int main()**

**{**

**int i = 10;**

**int j = i++ + ++i ;**

**printf("%d %d",i,j);**

**return 0;**

**}**

**Options:**

**a) 11 22**

**b) 11 21**

**c) 12 22**

**d) 12 21**

**9) #include<stdio.h>**

**#include<malloc.h>**

**int main()**

**{**

**int \*i , a = 9;**

**i=&a;**

**int j = \*i;**

**\*i=10;**

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

**return 0;**

**}**

**Options:**

**a) 10**

**b) 9**

**c) runtime error**

**d) garbage value**

**10)#include<stdio.h>**

**int main()**

**{**

**char ch;**

**for(ch=1;ch!=0;ch++)**

**printf("CodeWreck 1.0");**

**return 0;**

**}**

**Options:**

**a. Infinite loop.**

**b. Runs Perfectly.**

**c. Runtime Error.**

**d. Stack overflow.**

**11)#include<stdio.h>**

**int main()**

**{**

**int bike=10;**

**int cycle=70;**

**int auto=20;**

**int bus=10;**

**printf("%d",bike+cycle\*auto+bus);**

**return 0;**

**}**

**Options:**

**a) 1610**

**b) 1420**

**c) runtime error**

**d) compile error**

**12) #include<stdio.h>**

**int main()**

**{**

**int a=5;**

**if ( a & 1 )**

**printf("even");**

**else**

**printf("odd");**

**return 0;**

**}**

**Options:**

**a) odd**

**b) even**

**c) Compiler error**

**d) oddeven**

**13) #include<stdio.h>**

**void fun()**

**{**

**#define printf(x, y) printf(x, 0);**

**}**

**int main()**

**{**

**int j;**

**for( j=0;j<3;j++)**

**{**

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

**fun();**

**}**

**return 0;**

**}**

**Options:**

**a) 0 0 0**

**b) 0 1 2**

**c) Compiler error**

**d) Runtime error**

**14) #include<stdio.h>**

**int main()**

**{**

**char \* str = "CodeWreck";**

**int \* p, arr[] = { 4,3,2,1 };**

**p = &arr;**

**str++;**

**p++;**

**printf("%d %s",\*p,str);**

**return 0;**

**}**

**Options:**

**a) 3 odeWreck**

**b) 5 DodeWreck**

**c) 5 odeWreck**

**d) 3 DodeWreck**

**15) #include <stdio.h>**

**typedef struct s1**

**{**

**int a;**

**float b;**

**union g1**

**{**

**char c;**

**double d;**

**};**

**} new1;**

**int main()**

**{**

**printf("%d ",sizeof(new1));**

**return 0;**

**}**

**Options:**

**a) 17**

**b) 16**

**c) 8**

**d) 9**

**16) #include <stdio.h>**

**int main()**

**{**

**int m=2 , i=0;**

**switch (m)**

**{**

**case 1:**

**i+=1;**

**break;**

**case 2:**

**i+=2;**

**case 3:**

**i+=3;**

**default:**

**i+=4;**

**i+=5;**

**}**

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

**return 0;**

**}**

**Options:**

**a) 2**

**b) 5**

**c) 10**

**d) 14**

**17) #include <stdio.h>**

**int main()**

**{**

**signed a;**

**signed b;**

**printf("The size of a+b is %d\n", sizeof(a)+sizeof(b));**

**return 0;**

**}**

**Note: Consider size of int =4, size of double=8, size of char= 1.**

**Options:**

**a. The size of a+b is 8**

**b. The size of a+b is 16**

**c. The size of a+b is 2**

**d. Error: Signed was not declared in this scope**

**18)**

**#include<stdio.h>**

**int main()**

**{**

**long double a=0.7;**

**if(a<0.7)**

**printf("CodeWreck");**

**else**

**printf("2k17");**

**return 0;**

**}**

**Options:**

**a. CodeWreck**

**b. 2k17**

**c. Codewreck2k17**

**d. Compile error**

**19) #include<stdio.h>**

**int main()**

**{**

**int a = 10;**

**printf("%d %d %d %d", a, a++, - - a,a - -);**

**return 0;**

**}**

**Options:**

**a. 9 8 9 10**

**b. 10 10 10 10**

**c. 10 10 9 9**

**d. compiler dependent**

**20) #include<stdio.h>**

**int main()**

**{**

**int a = 10,b=20;**

**a=a^b;**

**b=a^b;**

**a=a^b;**

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

**return 0;**

**}**

**Options:**

**a. 20 10**

**b. 10 20**

**c. 10 10**

**d. 20 20**

**21)**

**#include<stdio.h>**

**int main()**

**{**

**if(!printf("CodeWreck "))**

**printf("2k17");**

**else**

**printf("2017");**

**return 0;**

**}**

**Options:**

**a. CodeWreck 2k17**

**b. CodeWreck 2017**

**c. 2017**

**d. 2k17**

**22) Which of the following will return a result most quickly for searching a given key?**

**Options:**

**a) Unsorted Array + binary search**

**b) Sorted Array + binary search**

**c) binary tree + preorder traversal**

**d) Binary Search tree**

**23) #include <stdio.h>**

**void f(char\*\*);**

**int main()**

**{**

**char \*argv[] = { "ab", "cd", "ef", "gh", "ij", "kl" };**

**f(argv);**

**return 0;**

**}**

**void f(char \*\*p)**

**{**

**char \*t;**

**t = (p += sizeof(int))[-1];**

**printf("%s\n", t);**

**}**

**a. ab**

**b. cd**

**c. ef**

**d. gh**

**24) #include<stdio.h>**

**int main()**

**{**

**int \*p = 15;**

**printf("%d",\*p);**

**return 0;**

**}**

**Options:**

**a. 15**

**b. Garbage value**

**c. Runtime error**

**d. Compiler error**

**25) #include <stdio.h>**

**#define fun(a,b) b##a**

**int main()**

**{**

**int a=10 , ab = 20 , b = 30 , ba = 40;**

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

**return 0;**

**}**

**Options:**

**a. 10**

**b.40**

**c. 30**

**d. Compilation error**

**26)**

**What does the following function do?**

|  |
| --- |
| **int fun(unsigned int n)**  **{**  **if (n == 0 || n == 1)**  **return n;**    **if (n%3 != 0)**  **return 0;**  **return fun(n/3);**  **}** |

**Options:**

**a. It returns 1 when n is a multiple of 3, otherwise returns 0**

**b. It returns 1 when n is a power of 3, otherwise returns 0**

**c. It returns 0 when n is a multiple of 3, otherwise returns 1**

**d. It returns 0 when n is a power of 3, otherwise returns 1**

**27)**

|  |
| --- |
| **Consider the following recursive C function that takes two arguments.**  **unsigned int foo(unsigned int n, unsigned int r)**  **{**  **if (n > 0)**  **return (n%r + foo (n/r, r ));**  **else**  **return 0;**  **}** |
| **What is the return value of the function foo when it is called as foo(345, 10) ?**  **a. 345**  **b. 12**  **c. 5**  **d. 3** |

**28)**

**int f(int n)**

**{**

**static int i = 1;**

**if (n >= 5)**

**return n;**

**n = n+i;**

**i++;**

**return f(n);**

**}**

**What will be the answer for f(1)?**

**Options:**

**a. 5**

**b. 6**

**c. 7**

**d. 8**

**29) What will be the output of program?**

**#include<stdio.h>**

**int main()**

**{**

**printf("%d%d", printf("1"), printf("-1"));**

**return 0;**

**}**

**Options:**

**a. 1-111**

**b. 1-112**

**c. -1112**

**d. 121-1**

**30)**

**You have to sort 1 GB of data with only 100 MB of available main memory. Which sorting technique will be best approach ?**

**a. Quick Sort**

**b. Merge Sort**

**c. Insertion Sort**

**d. Heap Sort**

**Answer Your Questions here:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| **6** | **7** | **8** | **9** | **10** |
| **11** | **12** | **13** | **14** | **15** |
| **16** | **17** | **18** | **19** | **20** |
| **21** | **22** | **23** | **24** | **25** |
| **26** | **27** | **28** | **29** | **30** |