|  |  |
| --- | --- |
| 4 rounds – Aptitude, Technical, HR, Behavioral | |
| **Technical** | **Answers** |
| 1) int main()  {  int m=10;  int x=printf("%d ",m);  printf("%d",x);  return 0;  }  A.10 3 B.103 C.10 2 D.102 | **A.10 3**  printf() returns total number of printed characters, the statement int x=printf("%d ",m) will print 10 (10 and one space) and return 3. Thus output will be 10 3 [10<space>3]. |
| 2) #include <stdio.h>  void main()  {  unsigned char c=290;  printf("%d",c);  }  A.34 B.290 C.Garbage D.45 | **A. 34**  290 is beyond the range of unsigned char. Its corresponding value printed is: (290 % (UINT\_MAX +1) where UINT\_MAX represents highest (maximum) value of UNIT type of variable.  Here it's character type and thus UINT\_MAX=255  Thus it prints 290 % (UINT\_MAX +1)=34 |
| 3) #define print "%sincludehelp "  int main()  {  int a=1,b=2,c=3;  printf(print,print);  return 0;  }  A. Garbage value  B. includehelp %sincludehelp  C. %sincludehelp includehelp  D. None of the above | **C. %sincludehelp includehelp**  After pre-processing phase of compilation, printf statement will become.  printf ("%sincludehelp ","%sincludehelp ");  It prints "%sincludehelp includehelp" |
| 4) #include <stdio.h>  void main()  {  int x=10;  x+=(x++)+(++x)+x;  printf("%d",x);  } | **45**  1) expand the expression : x=x+(x++)+(++x)+x;  2) due to pre increment ++x , x will be 11 for this expression.  3) after executing expression x will be 44.  4) finally x will be 45 due to post increment (x++). |
| 5) #include <stdio.h>  int main()  {  char var=0x04;  var = var | 0x04;  printf("%d,",var);  var = 5|2;  printf("%d",var);  return 0;  }  A. 8,9 B. 4,7  C. 8,7 D. 4,4 | **B. 4,7**  Value of var is 0x04 (0100), Consider the expression var = var | 0x04 The OR (|) of 0100, 0100 is 0100, hence value will remain 0100. After that it ORs the decimal number by converting them to hexa digits. |
| 6) #include <stdio.h>    int main()  {  int x=2.3;  const char c1=(float)x;  const char c2=(int)x;    printf("%d,%d\n",c1,c2);    return 0;  }  A.Error B.2.3,2 C.2.300000,2 D.2,2 | **D. 2.2**  const char can't be assigned to float casted value, thus assigning rounding of integer value. |
| 7) #include <stdio.h>  int main()  {  char \*str="IncludeHelp";  printf("%c\n",\*&\*str);  return 0;  }  A.Error B.IncludeHelp C. I D. \*I | **C. I**  & is a reference operator, \* is de-reference operator, We can use these operators any number of times. str points the first character of IncludeHelp, \*str points "I", \* & again reference and de-reference the value of str. |
| 8) #include <stdio.h>  int main()  {  char \*str[]= {"AAAAA","BBBBB","CCCCC","DDDDD"};  char \*\*sptr []={str+3,str+2,str+1,str};  char \*\*\*pp;    pp=sptr;  ++pp;  printf("%s",\*\*++pp+2);  return 0;  }  A. BBBBB B. CCCCC C.BBB D.Error | **C. BBB**  \*str is a array pointer of string, \*\*sptr is array pointer(double pointer) that is pointing to str strings in reverse order. \*\*\*pp also a pointer that is pointing sptr base address.  **++pp** will point to 1st index of sptr that contain str+2 ("CCCCC"). in**printf("%s",\*\*++pp+2);** ++pp will point to str+1, and \*\*++pp, value stored @ str+1 ("BBBBB). and **(\*\*++pp)+2** will point the 2nd index of "BBBBB", hence BBB will print. |
| 9) #include <iostream.h>  int a=10;  int main()  {  int a=20;  cout<<::a;  return 0;  }  A. 10 B. 20 C. ::10 D. ::20 | **A.10**  :: is a Scope Resolution Operator, it is used to access the value of global variable and ::a will return the value of global declared variable. |
| 10) #include <iostream>  using namespace std;  int main()  {  cout.operator<<("hello");  return 0;  }  A. hello  B. A hexadecimal address  C. Runtime Error  D. Linker error | **B. A hexadecimal address**  This operator symbol displays the address in the console screen. |

C or C++ or Java programming

1. Run lenth encoding
2. Program to swap without using temporary variable.
3. Program to print hello without using semicolon
4. Program to add numbers without using arithmetic operator
5. In an array 1-100 numbers are stored, one number is missing how do you find it?

And in technical round in the year of 2021

Self introduction, questions based on resume, basic questions from communication media, relays, programming languages and other basic things that we come across are asked.