**Final Exam Answer Script**

|  |
| --- |
| Question No. 01 |
| Question:  How many different ways are there to use functions? Explain with examples. |
| Answer:  There are four different ways to use functions:   1. Has Retuen and parameter   #include <stdio.h>  int add(int num1, int num2)  {  int sum= num1+num2;;  return sum;  }   1. No return and no parameter   #include <stdio.h>  void introduction()  {  printf("Hi\n");  }   1. No return but has parameter   #include <stdio.h>  void add(int num1, int num2)  {  int sum= num1+num2;;  printf(“%d”, sum);  }   1. Has return but no parameter   #include <stdio.h>  int add()  {  int num1,num2;  scanf(“%d%d”, &num1, &num2);  int sum= num1+num2;;  return sum;  } |

|  |
| --- |
| Question No. 02 |
| Question:  Take an integer N as input and write a for loop that prints all values from 1 to N that  are divided by 2 and 7 |
| Answer:  #include <stdio.h>  int main()  {  int n,a;  scanf("%d", &n);  for(int i=1;i<=n;i++){  scanf("%d", &a);  if(a%2==0 && a%7==0){  printf("%d\n",a);}    }    return 0;  } |

|  |
| --- |
| Question No. 03 |
| Question:  What are pointers? Give some examples of pointers. Make a pointer that points to a  structure variable (format is given below), and change the value of sum using  pointer.  struct Fun  {  int sum;  } |
| Answer:  A pointer is a variable that can stores another variables address in the memory location.  #include <stdio.h>  int main()  {  int c = 5;  printf("address of c: %p", &c);  return 0;  } |

|  |
| --- |
| Question No. 04 |
| Question:  Why do we need long long int? Give an example of a frequency array to count  integer values. Can you make a frequency array that counts long long int type  values? If the answer is yes, tell how? If the answer is no, tell why? |
| Answer:   The long long takes twice as much memory as long. If we use too long decimal number in our program, we need to use long long int, otherwise, it will be overflow. |

|  |
| --- |
| Question No. 05 |
| Question: |
| Answer: |

|  |
| --- |
| Question No. 06 |
| Question: |
| Answer: |

|  |
| --- |
| Question No. 07 |
| Question:.  Take an array name ar[] of size N where the values will be unique. Also take  another integer value named mul as input. Print “YES” if you can make mul by  multiplying two different values from that array. Otherwise, print “NO”. Here, (0 < N  <= 100 and 0 <= ar[i] <= N and 0 <= mul <= 10000) |
| Answer:  #include <stdio.h>  int main()  {  int i,n,m,sum=0;  scanf("%d", &n);  int ar[n];  for(i=0;i<n;i++){  scanf("%d", &ar[i]);  sum+=ar[i];    }  scanf("%d", &m);    if(sum==m)  printf("YES\n");  else  printf("NO\n");      return 0;  } |

|  |
| --- |
| Question No. 08 |
| Question:  Take an integer N as input and make a pattern for that number. Sample is given  below: |
| Answer:  #include<stdio.h>  int main()  {  int n, x, y;  scanf("%d",&n);  for(int x = 1; x < n; x++)  {  for(int y = 1; y <= x; y++)  printf("\*");  printf("\n");  }  for(int x = n; x >= 0; x--)  {  for(int y = 1; y <= x; y++)  printf("\*");  printf("\n");  }  return 0;  } |

|  |
| --- |
| Question No. 09 |
| Question:  What is structure in C? Make an example to create a structure named Student with  three integer values named roll, class, marks and make a variable with it.  Now make an array of that structure of size N which will take input from the user and  calculate the sum of marks of all students and print it. |
| Answer:  A struct in the is a composite data type (or record) declaration that defines a physically grouped list of variables under one name in a block of memory.  #include <stdio.h>  struct student {  char Name[50];  int roll;  int class;  float marks;  } s[5];  int main() {  int i;  for ( i = 0; i < 5; ++i) {  s[i].roll = i + 1;  printf("\nFor roll number%d,\n", s[i].roll);  printf("Enter first name: ");  scanf("%s", s[i].Name);  printf("Enter class: ");  scanf("%d", &s[i].class);  printf("Enter marks: ");  scanf("%f", &s[i].marks);  }  printf("Displaying Information:\n\n");  for (i = 0; i < 5; ++i) {  printf("\nRoll number: %d\n", i + 1);  printf("First name: ");  puts(s[i].Name);  printf("Marks: %d", s[i].class);  printf("Marks: %.1f", s[i].marks);  printf("\n");  }  return 0;  } |

|  |
| --- |
| Question No. 10 |
| Question:  It’s time to say Goodbye. But not to programming right? So, print “Goodbye” 100  times. Isn’t that so easy? No, you’re wrong. Print “Goodbye” 100 times, but if i-th  (where i means 1 to 100) term is an odd number then print “I Love Programming!” |
| Answer:  #include <stdio.h>  int main()  {  for(int i=1;i<=100;i++){  if(i%2==0)  printf("Goodbye\n");  else  printf("I Love Programming!\n");  }  return 0;  } |