**Question 1 [4 + 4 + 3 + 4 = 15 Marks]**

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

|  |  |  |
| --- | --- | --- |
| S no. | **Question** | **Output** |
| 1.1 | #include <iostream>  using namespace std;  int a = 21;  int b = 9;  int \*p = &a;    int\* func1(){  return &b;  }  int\* func2 (int\* p){  return p;  }  int& func2(){  return \*p;  }  int& func3(){  return a;  }  int main() {  int a = 8;  int\* p ;  cout << \*(func1()) << endl;    p = func2(&::a);  cout << \*p+5 << endl;    func2() = 1;  cout << ::a+4 << endl;    a= func3();  cout << a+2 <<endl;    return 0;  } | **9**  **26**  **5**  **3** |
| 1.2. | #include <iostream>  using namespace std;  int fun(int\* p , int s)  {  p = new int [s];  for (int i=0; i<s ; i++)  p[i] = (i+1)\*2;  return \*p;  }  int main() {  int \*p;  int x = fun(p,5);  for (int i=4 ; i>0; i--)  cout << x + i << endl;  return 0;  } | **6**  **5**  **4**  **3** |
| 1.3. | #include <iostream>  using namespace std;  int\* func(int \*pi)  {  int a = 7;  int \*\*p2 = &pi;  \*(\*p2) = a+5;  return \*p2;  }  int main(){  int r= 10;  int \*p = &r;  int\* x = func(p);  cout << (\*p) + (\*x);  return 0;  } | **24** |
| 1.4. | #include <iostream>  using namespace std;  int main(){  bool x = 0;  int y = 19;  int \*i= &y;  void \*ptr = &x;  ptr = &y;  cout << \*ptr;  const int\* p = &y;  (\*p)++;  cout << i;  p = new int(0);  cout << \*p;  return 0;  } | **Error 1: Cannot dereference void pointer**  **Error 2: P is a read only operator we cannot increment in its value** |

**Question 2 [ 4 + 3 + 6 + 4 = 17 Marks]**

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

|  |  |  |
| --- | --- | --- |
| S no. | **Question** | **Output** |
| 2.1. | #include <iostream>  using namespace std;  int MyFunction(int s, int t)  {  if (t != 0){  s--;  return (s\*MyFunction(s, t-1));  }  else{  return 1;  }  }  int main(){    cout<<"Ouput "<<MyFunction(5,3);    return 0;  } | **Output 24** |
| 2.2 | #include <iostream>  using namespace std;  int fun(int n, int\* fp)  {  int t, f;    if (n <= 2) {  \*fp = 1;  return 1;  }  t = fun(n - 1, fp);  f = t + \*fp;  \*fp = t;  return f;  }    int main()  {  int x = 15;  cout << fun(5, &x) << endl;  cout << x;  return 0;  } | **5**  **3** |
| 2.3 | #include <iostream>  using namespace std;  void print\_asterisk(int asterisk)  {  if (asterisk == 0)  return;  cout << "^ ";    print\_asterisk(asterisk - 1);  }  void print\_space(int space)  {  if (space == 0)  return;  cout << " "  << " ";    print\_space(space - 1);  }      void pattern(int n, int num)  {  if (n == 0)  return;    print\_asterisk(n);  print\_space(2 \* (num - n) + 1);  print\_asterisk(n);  cout << endl;  pattern(n - 1, num);  }    int main()  {  int n = 5;  pattern(n, n);  return 0;  } | **^ ^ ^ ^ ^ ^ ^ ^ ^ ^**  **^ ^ ^ ^ ^ ^ ^ ^**  **^ ^ ^ ^ ^ ^**  **^ ^ ^ ^**  **^ ^** |

**Question 2.4**

Write a recursive function called powerofFive. Given an integer n, return true if n is a power of five. Otherwise, return false.

An integer n is a power of five, if there exist an integer x such that n == 5x

Example1: Input n = 125

Output = true

Example1: Input n = 21

Output = false

|  |  |
| --- | --- |
| bool powerofFive(int n)  {      **if (n==1)**  **return true;**  **else if ( n < 5)**  **return false;**  **else**  **powerofFive(n/5);**        } | int main()  {    int n=125;    cout<< powerofFive (n); //print true  n=21;  cout<< powerofFive (n); //print false    return 0;    } |

**Question 3 [4+3+3=10 Marks]**

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

|  |  |  |
| --- | --- | --- |
| S no. | **Question** | **Output** |
| 3.1. | #include <iostream>  using namespace std;  int main(){  int arr[] = {-6,8,5,0,8,3,5,7,12};  int\* ptr;  ptr = arr;  arr[1] += 30;  cout << "Pointer>" << \*ptr << endl;  \*ptr -= 10;  ptr++;  cout << "Next" << ++(\*ptr) << endl;  ptr += 2;  cout << "Last" << ptr[0] << endl;  cout << "Final" << \*(arr+4)\*3 << endl;  return 0;  } | **Pointer>-6**  **Next39**  **Last0**  **Final24** |
| 3.2 | #include <iostream>  using namespace std;  int main()  {  const int x = 11;  const int \*const ptr = &x;  int y = 15;  const int \*p = &x;  int \* const ptr2 = &y;  ptr2 = p;  cout<<\*p<<" "<<\*ptr<<" "<<\*ptr2;  return 0;  } | **Error: ptr2 cannot point to any other variable except y because it’s a constant pointer.** |
| 3.3 | #include <iostream>  using namespace std;  int main()  {  float data[] = { 10.2,20.0, 30.5, 40.5, 76.1};  double \* a = new double;  \*a = \*(data + 2);  a = data;  \*a = (\*a - \*(a - 1));  cout << \*(data+3);  return 0;  } | **Error: pointer a is of double point. It cannot point to float type of array.** |

**Question 4 [8 Marks]**

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;    struct someStruct  {  int x = 1;  int y = 10;  int\*\* z;  someStruct\* ptr;    }s1 = { 3,10,0,0 };    int main() {  s1.z= new int\* [s1.x+1];  s1.z[0] = new int [4];  s1.z[1] = new int [1];  \*(s1.z[0]) = -90;  \*(s1.z[0] +1) = -12;  \*(s1.z[1]) = 3;  s1.ptr = new someStruct{ 2,86 };  s1.ptr->z = new int\* [s1.ptr->x];  s1.ptr->z[0] = new int [4];  s1.ptr->z[1] = new int [1];  \*(s1.ptr->z[0]) = 65;  \*(s1.ptr->z[0] + 1) = -2;  \*(s1.ptr->z[1]) = 87;    s1.ptr->ptr = &s1;  int i = 0, j=0;    do{  cout << \*(\*(s1.z+i)) << endl;  cout << \*(\*(s1.ptr->z+j)) << endl;  i++;  j++;  } while (i!=(s1.x-1));    return 0;  } | **-90**  **65**  **3**  **87** |

---------------------------------------------------------Rough Work-------------------------------------------------------