**CPP Problem Design**

|  |
| --- |
| **Contributor︰Wen-Kai, Wang** |
| **Subject：Combinations** |
| **Main testing concept：**   |  |  | | --- | --- | | **Basics** | **Functions** | | * C++ BASICS 1 * FLOW OF CONTROL * FUNCTION BASICS * PARAMETERS AND OVERLOADING * ARRAYS * STRUCTURES AND CLASSES * CONSTRUCTORS AND OTHER TOOLS * OPERATOR OVERLOADING, FRIENDS,AND REFERENCES * STRINGS * POINTERS AND DYNAMIC ARRAYS | * SEPARATE COMPILATION AND NAMESPACES * STREAMS AND FILE I/O * RECURSION * INHERITANCE * POLYMORPHISM AND VIRTUAL FUNCTIONS * TEMPLATES * LINKED DATA STRUCTURES * EXCEPTION HANDLING * STANDARD TEMPLATE LIBRARY * PATTERNS AND UML | |
| **Description：**  Please write a recursive function **PrintCombination** to enumerate all combinations of . Please copy the following code as your main function and do not change any code of it:   |  | | --- | | #define ELEMENTS\_FOR\_COMBINATION 5 //i.e., C(5,4)  #define DLEMENTS\_FOR\_CHOICE 4  int main(void)  {  int \*arrayPtr = new int[ELEMENTS\_FOR\_COMBINATION];  //Get all elements for combination  for (int i = 0; i < ELEMENTS\_FOR\_COMBINATION; ++i)  arrayPtr[i] = i + 1;  PrintCombination(arrayPtr, ELEMENTS\_FOR\_COMBINATION, DLEMENTS\_FOR\_CHOICE);  if (arrayPtr != NULL)  delete[] arrayPtr;  return 0;  } |   The function, PrintCombination(int \*, int, int), is for print all the combination results on console by recursive.  **Input：**no input  **Output：**  **Sample input / output :**   |  |  | | --- | --- | | **Sample Input** | **Sample Output** | | No input. | 1 2 3 4  1 2 3 5  1 2 4 5  1 3 4 5  2 3 4 5 | |

|  |
| --- |
| ■ Eazy,Only basic programming syntax and structure are required.  □ Medium,Multiple programming grammars and structures are required.  □ Hard,Need to use multiple program structures or complex data types. |
| **Expected solving time:**  20 minutes |
| **Other notes:** |