CHAPTER-8

POINTERS

VERY SHORT/SHORT ANSWER QUESTIONS

1.	**Describe C++ memory map when a program is executing. Highlight the role and use of Heap area.**
Ans.	When a program is executed, C++ creates four logically distinct regions of memory: (i) area to hold the compiled program code (ii) area to hold global variable (iii) the stack are to hold the return addresses of function calls, arguments passes to the functions, local variables for functions, and the current state of the CPU. (iv) the heap area from which the memory is dynamically allocated to the program.
2.	**What is pointer arithmetic? How is it performed? Support your answer with example.**
Ans.	Only two arithmetic operations, addition ad subtraction, may be performed on pointers. In pointer arithmetic, all pointers increase and decrease by the length of the data type they point to.For example, int *p; p++; If address of p is 1000, then p++ statement will increase p to 1002, not 1001.

3.	**Given the following definitions:**
	`int ival=2048; int *iptr; double *dptr;`
	which of the following assignment, if any, are illegal? Explain why.
	`(a) ival=*iptr (b) *iptr=ival;`
	`(c) *iptr=&ival (d) dptr=iptr;`
	`(e) ival=iptr; (f) iptr=ival;'`
	`(g) iptr=&ival; (h) dptr=*iptr;`

Ans.	(a)legal assignment. (b) legal assignment. (c)illegal assignment, cannot assign a address of normal variable to pointer variable. (d) illegal assignment, cannot assign a integer pointer to a double pointer. (e) illegal assignment, cannot assign pointer variable to a normal variable. (f) illegal assignment, cannot assign normal variable to a pointer variable. (g) legal assignment. (h) illegal assignment, cannot assign a integer pointer to a double pointer.

4.	**Given the following set of variable definitions:**
	`int *ip1,ip2;`
	`char ch,*cp;`
	which of the following assignment are type violations? Explain why.
	`(a) ip1="Smile Always"; (b) cp=0;`
	`(c) ip1=0; (d) cp=&'a';`
	`(e) ip1=ip2; (f) cp='\0';`
	`(g) ip1='\0'; (h) cp=&ch;`
	`(i) *ip=ip2;`

Ans.	(a) Type violation, cannot assign string to integer pointer. (b) Correct (c) Correct (d) Type violation, cannot assign string with '&' operator. (e) Type violation, cannot assign normal variable to a pointer variable. (f) Correct (g) Correct (h) Correct (i) Type violation, variable '*ip' is undefined.

5.	**What is the problem with the following code fragment? How might you fix it?**
	`char buf[]="Hi there";`
	`int main()`

	`{ char *ptr;`
	`for(int i=0;buf[i]!='\0';++i)`
	`ptr[i]=buf[i];`
	`return 0;`
	`}`

Ans.	There is no problem in above code.

6.	**What will be the output of the following program? Give proper explanation for this output.**
	`#include<iostream.h>`
	`int main()`
	`{ int a,*b,**c,***d;`
	`a=12;`
	`b=&a;`
	`c=&b;`
	`d=&c;`
	`cout<<a<<a+*b<<"\n";`
	`cout<<**c+***d<<"\n";`
	`return 0;`
	`}`

Ans.	Output: 1224 24 Explanation: While executing the program, it prints value of variable 'a' which is 12. After that it prints a+*b which is 24 as 'b' stores the address of 'a', and at last it prints **c+**d which is again 24 as 'd' store the address of 'c' and 'c' store the address of 'b'.

7.	**Predict and explain the output of the following program:**
	`#include<iostream.h>`
	`void swap(int *,int *);`
	`void exchange(int **,int *);`
	`int main()`
	`{ int a=5,b=15;`
	`cout<<"Before Swap, a="<<a<<"and b="<<b<<"\n";`
	`swap(&a,&b);`
	`cout<<"\n After Swap, a="<<a<<"and b="<<b<<"\n";`
	`return 0;`
	`}`
	`void swap(int *aa,int *bb)`
	`{ exchange(&aa,bb);`
	`}`
	`void exchange(int **cc,int *dd)`
	`{ int t;`
	`t=**cc;`
	`**cc=*dd;`
	`*dd=t;`
	`}`

Ans.	Output: Before Swap, a=5 and b=15 After Swap, a=15 and b=5 Explanation: The program contains two functions namely exchange and swap which call the exchange function. In main function swap function is called, which actually call the exchange function. In call of exchange function first argument is a address of variable and second argument is value of the variable. In call of the swap function both argument contains the address of the variable. After the execution of program the value of 'a' is 15 and value of 'b' is 5.
8.	**Predict and explain the output of the following program:** `#include<iostream.h>`

	`void change(int *);`
	`int main()`
	`{ int a[5]={4,5,6,7,8};`
	`change(a);`
	`for(int i=4;i>=0;i--)`
	`cout<<a[i];`
	`cout<<"\n";`
	`return 0;`
	`}`
	`void change(int *b)`
	`{ for(int i=0;i<=4;i++)`
	`{ *b=*b+1;`
	`b++;`
	`}`
	`}`

Ans.	Output: 98765 Explanation: This program contains one function called change() having one argument which is integer pointer. A change() function increment the pointer variable 'b' by 1. In main() function change() function is called and prints the array element from end which is incremented by 1 in change() function that is 98765.

9.	**Predict and explain the output of the following program:**
	`#include<iostream.h>`
	`int main()`
	`{ int iarr[]={10,12,14,16,18};`
	`int *ip;`
	`for(ip=&iarr[0];ip<=&iarr[4];ip++)`
	`cout<<*ip;`
	`cout<<"\n";`
	`return 0;`
	`}`

Ans.	Output: 1012141618 Explanation: This program contains integer pointer *ip which is initialized by address of first array element in for loop. A for is continued until variable 'ip' is less than or equal to address of last array element and prints actual value of array elements that is 1012141618.

10.	**Predict and explain the output of the following program:**
	`#include<iostream.h>`
	`int main()`
	`{ int iarr[]={10,11,12.13.14};`
	`int i,*p;`
	`for(p=iarr,i=0;p+i<=iarr+4;p++,i++)`
	`cout<<*(p+i);`
	`cout<<"\n";`
	`return 0;`
	`}`

Ans.	Output: 101214 Explanation: This program contains one integer variable 'i' and integer pointer '*p'. In for loop 'p' is initialized by integer array 'iarr' and 'i' is initialized with 0. A for loop is continued until 'p+i' is less than or equal to 'iarr+4' and prints addition of pointer variable 'p' and integer variable 'i' that is 10,12,14.

11.	**Find the output of the following program:**
	`#include<iostream.h>`
	`int main()`
	`{ int *Queen,Moves[]={11,22,33,44};`

```
      Queen=Moves;
      Moves[2]+=22;
      cout<<"Queen @"<<*Queen<<endl;
      *Queen-=11;
      Queen+=2;
      cout<<"Next @"<<*Queen<<endl;
      Queen++;
      cout<<"Finally @"<<*Queen<<endl;
      cout<<"New Origin @"<<Moves[0]<<endl;
}
```

Ans.

Output:
Queen @11
Next @55
Finally @44
New Origin @0

12. **Find the output of the following program:**

```
#include<iostream.h>
int main()
{     int Numbers[]={2,4,8,10};
      int *ptr=Numbers;
      for(int C=0;C<3;C++)
      {     cout<<*ptr<<"@";
            ptr++;
      }
      cout<<endl;
      for(C=0;C<4;C++)
      {     (*ptr)*=2;
            --ptr;
      }
      for(C=0;C<4;C++)
            cout<<Numbers[C]<<"#";
      cout<<endl;
      return 0;
}
```

Ans.

Output:
2@4@8@
4#8#16#20#

13. **Find the output of the following program:**

```
#include<iostream.h>
#include<string.h>
class state
{     char *state_name;
      int size;
     public:
      state()
      {     size=0;state_name=new char[size+1];
      }
      state(char *s)
      {     size=strlen(s);
            state_name=new char[size+1];
            strcpy(state_name,s);
      }
      void display()
      {     cout<<state_name<<endl; }
```

```
        void Replace(state &a,state &b)
        {     size=a.size+b.size;
              delete state_name;
              state_name=new char[size+1];
              strcpy(state_name,a.state_name);
              strcpy(state_name,b.state_name);
        }
};
int main()
{     char *temp="Delhi";
      state state1(temp),state2("Mumbai"),state3("Nagpur"),S1,S2;
      S1.Replace(state1,state2);
      S2.Replace(S1,state3);
      S1.display();
      S2.display();
      return 0;
}
```

Ans. Output:
Mumbai
Nagpur

14. **Find the output of the following program:**

```
#include<iostream.h>
#include<string.h>
class student
{     char *name;
      int I;
     public:
      student() {  I=0;name=new char[I+1]; }
      student(char *s)
      {     I=strlen(s); name=new char[I+1];
            strcpy(name.s);
      }
      void display() { cout<<name<<endl; }
      void manipulate(student &a,student &b)
      {     I=a.I+b.I;
            delete name;
            name=new char[I+1];
            strcpy(name,a.name);
            strcpy(name,b.name);
      }
};
int main()
{     char *temp="Jack";
      student name1(temp),name2("Jill"),name3("John"),S1,S2;
      S1.manipulate(name1,name2);
      S2.manipulate (S1,name3);
      S1.display();
      S2.display();
      return 0;
}
```

Ans. Output:
Jill
John

15. **Why is it said that it is faster to use an element pointer rather than an index when scanning arrays? Support your answer with an example.**

Ans.

The index version would:

```
int i = 0;
while( i < 100 ) {
  int* temp_ptr = number_seq + i;    // equivalent to:
  *temp_ptr = i;                     // number_seq[i] = i;
  i += 1;      // or ++i
};
```

While the pointer version is:

```
int* p = number_seq;
while( p != number_seq + 100 ) {
  *p = (p - number_seq);
  p += 1;    // or ++p
};
```

The index version adds one line of code, which is to compute the pointer to the element you are accessing before dereferencing that pointer. But most compilers will be able to see that the two lines (increment index, and add index to start-pointer) can be replace by just incrementing the pointer, which is why it ends up performing the same.

16. **Write a function that takes one string argument and returns a reversed string.**

Ans.

```
#include<iostream.h>
int string_length(char*);
void reverse(char*);
void main()
{  char string[100];
   cout<<"Enter a string\n";
   gets(string);
   reverse(string);
   cout<<"Reverse of entered string is \n"<< string;
   return 0;
}
 void reverse(char *string)
{  int length, c;
   char *begin, *end, temp;
   length = string_length(string);
   begin = string;
   end = string;
   for ( c = 0 ; c < ( length - 1 ) ; c++ )
      end++;
   for ( c = 0 ; c < length/2 ; c++ )
   {  temp = *end;
      *end = *begin;
      *begin = temp;
      begin++;
      end--;   }
}
int string_length(char *pointer)
{  int c = 0;
   while( *(pointer+c) != '\0' )
      c++;
   return c;
}
```

17. **Write a function that takes one string argument and returns 1 if the given string is a palindrome otherwise returns 0. (Note: *Palindrome string reads the same from both sides. For example, madam, level, malayalam***

	***are palindromes*)**

Ans.	`#include<iostream.h>`
	`#include<process.h>`
	`#include<stdio.h>`
	`#include<string.h>`
	`void palindrome(char a[])`
	`{`
	`int l=strlen(a);`
	`char *a1,*b1;`
	`for(a1=a,b1=&a[l-1];a1<=b1;a1++,b1--)`
	`{`
	`if(*a1!=*b1)`
	`{`
	`cout<<"No,it is not a Palindrome:";`
	`exit(0);`
	`}`
	`}`
	`cout<<"Yes,it is a Palindrome:";`
	`}`
	`void main()`
	`{`
	`char c[30];`
	`cout<<"Enter the string:";`
	`gets(c);`
	`cout<<endl;`
	`palindrome(c);`
	`}`

18.	**What will be the output of the following program?** **`int main()`** **`{  char a[]="able was I ere I saw elba";`** **`  char *t,*s,*b;`** **`  s=a;`** **`  b=a+strlen(a)-1;`** **`  t=b;`** **`  while(s!=t)`** **`  {  cout<<*s;`** **`  s++;`** **`  cout<<*t;`** **`  t--;`** **`  }`** **`  return 0;`** **`}`**
Ans.	Output: `aabbllee wwaass II ee`
19.	**Write a function to encode a string that is passed to it. The string should get converted into an unrecognizable form.** **[Hint: You may change the ASCII code of the constituent character by adding or subtraction a constant number from the original ASCII value of the constituent characters.]**

Ans.	`#include<iostream.h>`
	`#include<conio.h>`
	`#include<string.h>`
	`#include<stdio.h>`
	`void encode(char str1[],int l);`
	`void main()`
	`{`

```
      char str[30];
      clrscr();
      cout<<"Enter an string:";
      gets(str);
      int len=strlen(str);
      encode(str,len);
      getch();
}
void encode(char str1[],int l)
{
      int i;
      for(i=0;i<l;i++)
      {
            str1[i]=str1[i]+10;
      }
      cout<<"\nThe encoded string is:"<<endl;
      for(i=0;i<l;i++)
      {
            cout<<str1[i];
      }
      cout<<"\nThe proper string is:"<<endl;
      for(i=0;i<l;i++)
      str1[i]=str1[i]-10;
      for(i=0;i<l;i++)
      cout<<str1[i];
}
```

20. **Predict and explain the output of the following program:**

```
#include<iostream.h>
int main()
{     float *retpt(float *r);
      float p=22.6,*q;
      q=&p;
      cout<<"q before call="<<q<<"\n";
      q=retpt(&p);
      return 0;
}
float *retpt(float *r)
{     r+=1;
      return(r);
}
```

(Assume that q before call is 4004.)

Ans. Output:

```
q before call=0x8f64fff2
```

Explanation: This program contains one function namely 'retpt()' which has floating pointer as argument and increment *r by 1. A main() function contains one float variable p which is initialized with 22.6 and floating pointer '*q' which is initialized with address of variable 'p'. After that it prints the value of q before call to the function retpt() which is `0x8f64fff2.`

21. **Write a function search() which scans a string from beginning is to end in search of a character. If the character is found it should return a pointer to the first occurrence of the given character in the string. If the given character is not found in the string, the function should return a NULL. The prototype of the function would be:**

```
            char *search(char *,char);
```

Ans.

```
#include<iostream.h>
#include<conio.h>
```

```
#include<string.h>
#include<stdio.h>
char *search(char *,char);
int len;
void main()
{
      char str[20],ch,*f;
      clrscr();
      cout<<"Enter a string:";
      gets(str);
      cout<<"Enter the character to be searched for:";
      cin>>ch;
      len=strlen(str);
      f=search(str,ch);
      if(f!=NULL)
            cout<<"Character is found.";
      else
            cout<<"Character is not found.";
      getch();
}
char *search(char *x,char d)
{
      int i;
      for(i=0;i<len;i++)
      {
            if(*x==d)
            return x;
            x++;
      }
      return NULL;
}
```

22. **Write a function to compress any given string such that the multiple blanks preset in it are eliminated. The function should return the compressed string.**

Ans.

```
#include<iostream.h>
void removeSpaces(char* s)
{
    char* source = s;
    char* dest = s;
     while(*source) {
        if(*source == ' ')
        {
            source++;
        }
       else
        {
            *dest++ = *source++;
        }
    }
    *dest = 0;
}
int main()
 {
    char input[50] = "I       like            milk";
    removeSpaces(input);
    cout<<input;
```

```
    return 0;
}
```

23 **Predict and explain the output of the following program:**

```
int main()
{    struct S{   char carr[10];
                  int a;
                 float b;
              }sa[2];
     //assume that the first structure begins at address 1004
     cout<<sa[0].carr<<*(&sa[0]).a<<*(&sa[0]).b<<"\n";
     cout<<sa[1].carr<<*(&sa[1]).a<<*(&sa[1]).b<<"\n";
     return 0;
}
```

Ans. **Students can you do this... Try...**

24. **C++ lets you pass a structure by value and it lets you pass the address of a structure. If sample is a structure variable, how would you pass it by value? How would you pass its address? Which method do you think is better?**

Ans. If we assume structure variable name as sample and function name as sample_function() then:

Pass a structure by value	**Pass a structure by address**

struct SAMPLE sample;
sample_function(**sample**);
 	
void sample_function(struct SAMPLE **sample**)
{
  :
  :
}
	struct SAMPLE sample;
	sample_function(**&sample**);
	
	void sample_function(struct SAMPLE ***sample**)
	{
	  :
	  :
	}

Both the method have their own importance, so as per the program requirement we can use any of them.

25. **Write a function having this prototype:**

```
int replace(char *s,char c1,char c2);
```

Have the function replace every occurrence of c1 in the string s with c2, ad have the function return the number of replacements it makes.

Ans.

```
int replace(char * str, char c1, char c2)
{
    int count = 0;
    while (*str)      // while not at end of string
    {
        if (*str == c1)
        {
            *str = c2;
            count++;
        }
        str++;        // advance to next character
    }
    return count;
}
```

26. **Given here is a structure defilation:**

```
struct Box {    char maker[21];
                float height,width,length,volume;
           };
```

Write a function that passes the address of a Box structure and that sets the volume member to the

	product of the other three dimensions.

Ans.	`#include<iostream.h>`
	`struct box`
	`{`
	`  char maker[40];`
	`  float height;`
	`  float width;`
	`  float length;`
	`  float volume;`
	`};`
	`box yours;`
	`float boxer_dimensions (const box * compute );`
	`void display_box (const box make);`
	`int main()`
	`{`
	`box yours ={"Apple", 10.0, 5.0,7.5};// to initialize a record`
	`char name[40];`
	`float dia;`
	`dia= boxer_dimensions (&yours); //function call`
	`yours.volume = dia;// assigning  results to volume member record`
	`display_box (yours);// function call`
	`return 0;`
	`}`
	`float  boxer_dimensions (const box *compute )// specs calls for passing address of the structure`
	`{`
	`float Vol;`
	`Vol= ((compute->height)*(compute->length)*(compute->width));`
	`//computation for volume here`
	`return Vol;`
	`}`
	`void display_box (const box make)/// display all results here`
	`{`
	`cout << make.maker <<endl;`
	`cout << make.height <<endl;`
	`cout << make.width <<endl;`
	`cout << make.length<< endl;`
	`cout << make.volume<<endl;`
	`}`

27.	**What is the advantage of passing arguments by reference? When and why is passing an object by reference preferable?**
Ans.	In passing arguments by reference method, the called function does not create its own copy of original values; rather, it refers to the original values only by different names i.e., the references. Thus the called function works with the original data and any change in the values gets reflected to the data. The passing an object by reference method is preferable in situations where we want the called function to work with the original object so that there is no need to create and destroy the copy of it.
28.	**Write a function fibonacci() that takes an int argument n and prints first n elements of Fibonacci series. (A function series is where each successive number is the sum of preceding two elements. For example, 0 1 1 2 3 5 8 ….. is a Fibonacci series)**

Ans.	`#include<iostream.h>`
	`#include<stdio.h>`
	`#include<conio.h>`
	`int Fibonacci(int);`
	`void main()`

```
{
   int n, i = 0, c;
   clrscr();
   cout<<"enter number :";
   cin>>n;
   cout<<"Fibonacci series\n";
   for ( c = 1 ; c <= n ; c++ )
   {
      cout<<Fibonacci(i)<<endl;
      i++;
   }
   getch();
}
int Fibonacci(int n)
{
   if ( n == 0 )
      return 0;
   else if ( n == 1 )
      return 1;
   else
      return ( Fibonacci(n-1) + Fibonacci(n-2) );
}
```

29. **Correct the errors in the following program**

```
class Sample { private:
                int x;
               public:
                void getdata(void)
                {     cout<<"Enter number: ";
                       cin>>x;
                }
                void display(void)
                {     cout<<x;
                }
            };
int main()
{     Sample S;
      S->getdata();
      S->display();
      Sample *p;
      p=new Sample;
      p.getdata();
      (*p).display();
      return 0;
}
```

Ans.

```
class Sample { private:
                int x;
               public:
                void getdata(void)
                {     cout<<"Enter number: ";
                       cin>>x;
                }
                void display(void)
                {     cout<<x;
                }
            };
```

```
int main()
{     Sample S;
      S.getdata();
      S.display();
      Sample *p;
      p=new Sample;
      p->getdata();
      (*p).display();
      return 0;
}
```

30. **What will be the output of the following program? Run this program and find out:**

```
#include<iostream.h>
struct Book {    char *name;
                 char *author;
                 float royalty;
            };
int main()
{     Book b1={"Computers","DBA",10.5};
      Book *p;
      p=&b1;
      cout<<*p->name<<*p->author<<p->royalty<<"\n";
      cout<<p->name<<p->author<<p->royalty<<"\n";
      return 0;
}
```

Ans. Output:

CD10.5

ComputerDBA10.5

31. **Write a function that takes two string arguments and return the concatenated string.**

Ans.

```
#include<iostream.h>
void stcat(char *str1, char *str2);
void main()
{
char *str1, *str2;
clrscr();
cout<<"\n\n\t ENTER THE FIRST STRING: ";
gets(str1);
cout<<"\n\n\t ENTER THE FIRST STRING: ";
gets(str2);
stcat(str1,str2);
cout<<"\n\t THE CONCATENATED STRING IS: ";
puts(str1);
getch();
}
void stcat (char *str1, char *str2)
{
int i = 0,len = 0;
while(*(str1+len)!='\0')
len++;
while(*(str2+i)!='\0')
{
*(str1+len) = *(str2+i);
i++;
len++;
}
*(str1+len) = '\0';
```

	} **32.** **What are this and *this? How are these two different?** **Ans.** this is a pointer to the object whose member function is being executed. It is a pointer to the object itself while *this is an object pointed by the pointer this.
33.	**What will be the output of the following program?** **#include<iostream.h>** **int main()** **{ int x[3][5]={ {18,20,13,24,35},** **{7,8,6,19,10},** **{19,22,30,21,15} };** **int *n=&x[0][0];** **cout<<"1.(*(n+3)+1) \t="<<(*(n+3)+1)<<endl;** **cout<<"2. *(n+2) \t="<<*(n+2)<<endl;** **cout<<"3. *(*x+2)+5 \t="<<*(*x+2)+5<<endl;** **cout<<"4. ++*n \t="<<++*n<<endl;** **cout<<"5. *(*(x)+2)+1 \t="<<*(*(x)+2)+1<<endl;** **cout<<"6. *n \t\t="<<*n<<endl;** **cout<<"7. *(*(x+2)+1) \t="<<*(*(x+2)+1)<<endl;** **cout<<"8. *(*(x+1)+3) \t="<<*(*(x+1)+3)<<endl;** **cout<<"9. *(*(x+1)) \t="<<*(*(x+1))<<endl;** **cout<<"10. *(n+5)+1 \t="<<*(n+5)+1<<endl;** **return 0;** **}**
Ans.	Output: 1.(*(n+3)+1) = 25 2. *(n+2) = 13 3. *(*x+2)+5 = 18 4. ++*n = 19 5. *(*(x)+2)+1 = 14 6. *n = 19 7. *(*(x+2)+1) = 22 8. *(*(x+1)+3) = 19 9. *(*(x+1)) = 7 10. *(n+5)+1 = 8
34.	**What will be the output of the following program?** **#include<iostream.h>** **#include<stdlib.h>** **int main()** **{ system("cls");** **char string[]="Pointers and Strings";** **cout<<*(&string[2]<<'\n";** **cout.write(string+5,15).put('\n');** **cout.write(string,20).put('\n');** **cout<<*(string+3)<<"\n";** **return 0;** **}**
Ans.	Output: i ers and Strings Pointers and Strings n

LONG ANSWER QUESTIONS

1. **Write a function substr() that will scan a string for the occurrence of a give substring. the prototype of the function would be:**

```
char *substr(char *string1,char string2);
```

The function should return a pointer to the element in string1 where strig2 begins. If string2 doesn't occur in string1 then substr() should return a NULL.
For example, if string1 is "Ena Meena Deeka", and string2 is "Meena" the substr() should return address of 'M' in string1.
Write main() function also that uses the function substr().

Ans.

```
#include <iostream.h>
#include<stdio.h>
char *get_substr(char *sub, char *str);
int main()
{
  char *substr,*str;
  cout<<"Enter the string :";
  gets(str);
  cout<<"\nEnter the substring :";
  gets(substr);
  substr = get_substr(substr,str);
  if (substr!='\0')
  cout << "substring found: " <<substr;
  else
  cout<< "substring not found";
  return 0;
}

char *get_substr(char *sub, char *str)  // Return pointer to substring or
null if not found.
{
  int t;
  char *p, *p2, *start;

  for(t=0; str[t]!='\0'; t++)
   {
    p = &str[t];
    start = p;
    p2 = sub;
    while(*p2!='\0' && *p2==*p)       // check for substring
    {
      p++;
      p2++;
    }
    /* If at end of p2 (i.e., substring), then a match has been found. */
    if(!*p2)
      return start;         // return pointer to beginning of substring
  }
return 0;
}
```

2. **Suppose 7 names are stored in a array of pointers names[] as shown below:**

```
char *names[]= { "Anand","Naureen","Banjot","Wahid","sheena" };
```

Write a program to reverse the order of these names.

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
```

```
void main()
{
      clrscr();
      char *name[]={"anand","naureen","banjot","wahid","sheena"};
      int i,j;
      cout<<"\nOriginal string\n";
      for(i=0;i<5;i++)
      cout<<name[i]<<endl;
      char *t;
      for(i=0,j=4;i<5/2;i++,j--)
      {
            t=name[i];
            name[i]=name[j];
            name[j]=t;
      }
      cout<<"\nReversed string:\n";
      for(i=0;i<5;i++)
      cout<<name[i]<<endl;
      getch();
}
```

3. **Imagine a publishing company that markets both books and audio-cassette versions of its works. Create a class called Publication that stores the title (a string) and price of a publication. From this class derive two classes: Book, which adds a page count (type int); and Tape, which adds a playing time in minutes (type float). Each of the three class should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data.**
Write a main() program that creates an array of pointers to Publication. In a loop, ask the user for data about a particular book or Tape, and use new to create a object of type Book or Tape to hold the data. Put the pointer to the object in the data for all books and tapes, display the resulting data for all the books and taps entered, using a for loop and a single statement such as

```
pubarr[i]->putdata();
```

to display the data from each object in the array.

Ans.

```
#include<iostream.h>
#include<string.h>
class Publication
{
private:
    char title[20];
    float price;
public:
    void getName()
    {
     cout<<"Enter Title: "; cin>>title;
     cout<<"Enter Price: $"; cin>>price;
    }
    void putName()
    {
     cout<<"\nTitle: "<<title;
     cout<<", Price: $"<<price;
    }
    virtual void getData() = 0;
};
class Book : public Publication
{
private:
    int pages;
```

```
public:
    void getData()
    {
        Publication::getName();
     cout<<"Enter Pages: "; cin>>pages;
    }
    void putData()
    {
        Publication::putName();
     cout<<", Pages: "<<pages<<endl;
    }
};

class Tape : public Publication
{
private:
    float minutes;
public:
    void getData()
    {
        Publication::getName();
     cout<<"Enter Minutes: "; cin>>minutes;
    }
    void putData()
    {
        Publication::putName();
     cout<<", Minutes: "<<minutes<<endl;
    }
};
int main()
{
    Publication* ptrPub[100];
    int n = 0;
    char choice;
    do
    {
     cout<<"Book or Tape? (b/t): "; cin>>choice;
     if(choice == 'b')
        { ptrPub[n] = new Book; ptrPub[n]->getData(); }
        else
        { ptrPub[n] = new Tape; ptrPub[n]->getData(); }
     n++; cout<<"Enter another? (y/n): "; cin>>choice;
    } while(choice == 'y');

    for(int i=0; i<n; i++)
        ptrPub[i]->putName();
    cout<<endl;
    return 0;
}
```

4. **Given the following incomplete class definition:**

```
class Player
{    char firstname[15],surname[15];
     int runs,sixes,fours;
     float strike-rate;
    public:
```

	`Player(); //constructor`
	`~Player(); //Destruct`
	`char *getname();`
	`{ return strcat(firstname,lastname); }`
	`int getruns()`
	`{ rerurn runs; }`
	`int getsix()`
	`{ rerurn sixes; }`
	`int getfours()`
	`{ rerurn fours; }`
	`float getst_rate()`
	`{ rerurn strike-rate; }`
	`};`
	Write a program that uses array of pointers to object to store the details of various Players. Get details of the players from the user at the keyboard and print the details of the Player who has made maximum runs.

Ans.	
5.	**Create some objects of the Player class (of question 4), and put them on a List, an array of pointers to objects. Display the contents of the resulting list in two ways:** **(i) Use println() to print an entire list.** **(ii) Using a while loop ask for the Player's name and display his details and then remove this Player from the List.**
Ans.	**Students take help of answer number 4**