**Code :**

```
#include <iostream>
#include <ios>
#include <limits>
#include <fstream>
using namespace std;
struct Node
{
  string s = "";
  int position;
  Node *next;
  Node *prev;
} *root = NULL, *last = NULL,  *q = NULL, *r = NULL;
// undo
void undo()
{
  q = last->prev;
  r = last;
  last = q;
  cout << q->s << endl;
}
// redo
void redo()
{
  last = r;
  cout << last->s << endl;
  r = r->next;
}
// save
void save()
{
  ofstream myFile_Handler;
  myFile_Handler.open("TextFile.txt");
  myFile_Handler << last->s << endl;
  myFile_Handler.close();
  cout << "Saved the file..." << endl;
}
int main()
{
  int choice;
  begin:
  {
    cout << "Options:" << endl;
    cout << "1. Insert text" << endl;
    cout << "2. Move cursor" << endl;
    cout << "3. Move cursor to end" << endl;
```

```
  cout << "4. Backspace" << endl;
  cout << "5. Delete" << endl;
  cout << "6. Undo" << endl;
  cout << "7. Redo" << endl;
  cout << "8. Save" << endl;
  cout << "9. Quit" << endl;
  cout << "Enter your choice:" << endl;
  cin >> choice;
  cin.ignore(numeric_limits<streamsize>::max(), '\n');
}
switch (choice)
{
  case 1: // inserting text
  {
    bool q = true;
    do
    {
      char t[10000];
      cout << "Enter the text:" << endl;
      cin.getline(t,10000);
      if (root == NULL)
      {
        root = new Node;
        root->next = NULL;
        root->prev = NULL;
        root->s = t;
        root->position = root->s.length() - 1;
        last = root;
        cout << "Enter '1' to contiue or '0' to quit" << endl;
        cin >> q;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
      }
      else
      {
        Node *p = new Node;
        int l = last->s.length() - 1;
        if (last->position == l)
        {
          if ((last->s[l]) == ' ')
           p->s = last->s + t;
          else
           p->s = last->s + " " + t;
          p->position = p->s.length() - 1;
        }
        else
        {
          p->s.append(last->s, 0, last->position);
          p->s += t;
          p->s.append(last->s, last->position, l + 1);
          p->position = p->s.length() - 2 - (l - last->position);
```

```
        }
        p->next = NULL;
        p->prev = last;
        last->next = p;
        last = p;
        cout << "Enter '1' to continue or '0' to quit"<<endl;
        cin >> q;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
      }
    } while (q);
    cout << "Entered text is:" << endl;
    cout << last->s << endl;
    goto begin;
    break;
  }
  case 2: // moving cursor to specified position
  {
    if (root == NULL || last->position == 0)
      break;
    else
    {
     cout << "Enter position to move:" << endl;
     int position;
     cin >> position;
     if (position > last->s.length() - 1)
       break;
     Node *p = new Node;
     p->s = last->s;
     p->position = position;
     p->next = NULL;
     p->prev = last;
     last->next = p;
     last = p;
    }
    goto begin;
    break;
  }
  case 3: // moving cursor to the end
  {
    if (root == NULL)
      break;
    Node *p = new Node;
    p->s = last->s;
    p->position = last->s.length() - 1;
    p->next = NULL;
    p->prev = last;
    last->next = p;
    last = p;
    goto begin;
    break;
```

```
}
case 4: // backspace
{
  if (root == NULL)
   break;
  Node *p = new Node;
  p->s.append(last->s, 0, last->position );
  p->s.append(last->s, last->position+1, last->s.length());
  p->position = last->position - 1;
  p->next = NULL;
  p->prev = last;
  last->next = p;
  last = p;
  cout << last->s << endl;
  goto begin;
  break;
}
case 5: // deleting entire text
{
  if (root == NULL)
   break;
  Node *p = new Node;
  p->s = "";
  p->position = 0;
  p->next = NULL;
  p->prev = last;
  last->next = p;
  last = p;
  cout << last->s << endl;
  goto begin;
  break;
}
case 6: // undo
{
  undo();
  goto begin;
  break;
}
case 7: // redo
{
  redo();
  goto begin;
  break;
}
case 8: // save
{
  save();
  goto begin;
  break;
}
```

```
    case 9: // quit
      cout << last->s << endl;
      break;
  }
  return 0;
}
```