

# **Multimedia Lab File ITC15**

Submitted by : Shiv Kumar  
IT-2  
5th Semester  
2016UIT2563

# INDEX

1. WAP to write and display the data in file. (File handling in C)
2. WAP to compress and decompress the text using run length encoding technique.
3. WAP for text compression by Static Huffman coding in C.
4. WAP for text compression by Dynamic Huffman coding in C.
5. WAP for text compression by Arithmetic coding in C.
6. WAP for text compression by LZW compression technique in C
7. WAP to create a Keyboard Piano.
8. Create a web page to design your resume using basic HTML formatting Tags.
9. Create a web page to design a time table of your college for all the semesters and branch using anchor tag.
10. WAP to simulate the Sudoku puzzle in C.

# PROGRAM 1

**WAP to write and display the data in file. (File handling)**

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ofstream ofile;
    string s;
    cout << "Enter Data In File: ";
    getline(cin, s);
    ofile.open("ques1 TextFile.txt");
    ofile << s;
    ofile.close();
    ifstream infile;
    infile.open("ques1 TextFile.txt");
    char t[100];

    if(infile.fail())
    {
        cout << "file not Present!!!" << endl;
        return 0;
    }

    while (!infile.eof())
    {
        infile.getline(t, 100);
        cout << t << endl;
    }

    infile.close();
    return 0;
}
```

```
[Shivs-MacBook-Air:multimedia championballer$ g++ prac1.cpp -o run ]
[Shivs-MacBook-Air:multimedia championballer$ ./run ]
Enter Data In File: Hello nerd
Hello nerd
[Shivs-MacBook-Air:multimedia championballer$ ls ]
LIST.md          prac2.cpp        ques1TextFile.txt
prac1.cpp        prac3.cpp        run
```

# PROGRAM 2

**WAP to compress and decompress the text using run length encoding technique.**

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
string encode(string s)
```

```
{
```

```
    string ans = "";
```

```
    for (int i = 0; i < s.length(); i++)
```

```
    {
```

```
        int c = 1;
```

```
        while (i < s.length() - 1 && s[i + 1] == s[i])
```

```
        {
```

```
            c += 1;
```

```
            i++;
```

```
        }
```

```
        ans += s[i];
```

```
        ans += to_string(c);
```

```
    }
```

```
    return ans;
```

```
}
```

```
string decode(string s)
```

```
{
```

```
    string ans = "";
```

```
    for (int i = 0; i < s.length(); i += 2)
```

```
    for (int j = 0; j < s[i + 1] - 48; j++)
```

```
    ans += s[i];
```

```
    return ans;
```

```
}
```

```
int main()
```

```
{
```

```

string s;
cout << "Enter the string to be encoded :";
cin >> s;
string encoded = encode(s);
cout << "Encoded String for \"\" << s << "\"" is :\" << encoded << endl;
string decoded = decode(encoded);
cout << "Decoded String for \"\" << encoded << "\"" is :\" << decoded<<
endl;
return 0;
}

```

```

[Shivs-MacBook-Air:multimedia championballer$ g++ prac2.cpp -o run
[Shivs-MacBook-Air:multimedia championballer$ ./run
Enter the string to be encoded :wwwwaaaaddaa
Encoded String for "wwwwaaaaddaa" is :w3a4d2a2
Decoded String for "w3a4d2a2" is :wwwwaaaaddaa

```

# PROGRAM 3

## WAP for text compression by Static Huffman coding in C.

```
#include <bits/stdc++.h>
using namespace std;

struct Node
{
    char ch;
    int freq;
    Node *left, *right;
};

Node *getNode(char ch, int freq, Node *left, Node *right)
{
    Node *node = new Node();
    node->ch = ch;
    node->freq = freq;
    node->left = left;
    node->right = right;
    return node;
}

struct comp
{
    bool operator()(Node *l, Node *r)
    {
        return l->freq > r->freq;
    }
};

void encode(Node *root, string str, unordered_map<char,
    string>&huffmanCode)
{
    if(root == NULL)
        return;
    if(!root->left && !root->right)
        huffmanCode[root->ch] = str;
    encode(root->left, str + "0", huffmanCode);
```

```
    encode(root->right, str + "1", huffmanCode);  
}
```

```
void decode(Node *root, int &index, string str)  
{  
    if(root == NULL)  
        return;  
    if(!root->left && !root->right)  
    {  
        cout << root->ch;  
        return;  
    }  
  
    index += 1;  
    if (str[index] == '0')  
        decode(root->left, index, str);  
    else  
        decode(root->right, index, str);  
}
```

```
void buildHuffmanTree(string text)  
{  
    unordered_map<char, int> freq;  
    for (char ch : text)  
    {  
        freq[ch] += 1;  
    }  
}
```

```
priority_queue<Node *, vector<Node *>, comp> pq;  
for (auto pair : freq)  
{  
    pq.push(getNode(pair.first, pair.second, NULL, NULL));  
}
```

```
while (pq.size() != 1)  
{  
    Node *left = pq.top();  
    pq.pop();  
    Node *right = pq.top();  
    pq.pop();
```



```

        int sum = left->freq + right->freq;
        pq.push(getNode('\0', sum, left, right));
    }

```

```

Node *root = pq.top();
unordered_map<char, string> huffmanCode;
encode(root, "", huffmanCode);
cout << "Huffman Codes are:\n\n";
for (auto pair : huffmanCode)
{
    cout << pair.first << " " << pair.second << endl;
}

```

```

cout << "\nOriginal String was: " << text << "\n";
string str = "";
for (char ch : text)
{
    str += huffmanCode[ch];
}

```

```

cout << "\nEncoded String is: " << str << endl;
int index = -1;
cout << "Decoded string is: ";
while (index < (int)str.size() - 2)
{
    decode(root, index, str);
}
cout << endl;
}

```

```

int main()
{
    string s;
    cin >> s;
    buildHuffmanTree(s);
    return 0;
}

```

```
[Shivs-MacBook-Air:multimedia championballer$ g++ -std=c++11 prac3.cpp -o run ]
[Shivs-MacBook-Air:multimedia championballer$ ./run ]
waddawabba
Huffffman Codes are:

a 11
w 10
d 01
b 00

Original String was: waddawabba

Encoded String is: 10110101111011000011
Decoded string is: waddawabba
```

# Practical 4

## WAP for text compression by Dynamic Huffman coding in C.

```
#include <bits/stdc++.h>
using namespace std;

typedef struct node
{
    int freq;
    char data;
    struct node *left,*right;
}node;
node *newnode(char data,int freq)
{
    node *p=new node;
    p->data=data;
    p->freq = freq;
    p->left=p->right=NULL;
    return p;
}
void printArr(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        cout<<arr[i]<<" ";
    cout<<"\n";
}
void printCodes(node *root, int arr[], int top)
{
    if (root->left)
    {
        arr[top] = 0;
        printCodes(root->left, arr, top + 1);
    }
    if (root->right)
    {
        arr[top] = 1;
        printCodes(root->right, arr, top + 1);
    }
}
```

```

if (!root->right && !root->left)
{
    cout<<root->data<<" ";
    printArr(arr, top);
}
}
int height(node *root)
{
    if(root==NULL)
        return 0;
    return 1 + max(height(root->left),height(root->right));
}
void search(node *root,char ch,node *&p)
{
    if(root==NULL)
        return;
    if(root->data==ch)
        p=root;
    search(root->left,ch,p);
    search(root->right,ch,p);
}
int updatefreq(node *root)
{
    if(root==NULL)
        return 0;
    if(root->left==NULL && root->right==NULL)
        return root->freq;
    root->freq = updatefreq(root->left) + updatefreq(root->right);
    return root->freq;
}
void swapnodedata(node *a,node *b)
{
    node *tempn;
    char tempc;
    int tempf;
    //char
    tempc=a->data;
    a->data=b->data;
    b->data=tempc;
    //freq;

```

```

tempf=a->freq;
a->freq=b->freq;
b->freq=tempf;
//left
tempn=a->left;
a->left=b->left;
b->left=tempn;
//right
tempn=a->right;
a->right=b->right;
b->right=tempn;
}
bool anomalyswap(vector<node*> &rlo)
{
    int n=(int)rlo.size(),i,j;
    for(i=0;i<n;i++)
    {
        for(j=n-1;j>i;j--)
        {
            if(rlo[i]->freq>rlo[j]->freq)
            {
                swapnodedata(rlo[i],rlo[j]);
                return true;
            }
        }
    }
    return false;
}
bool reverselevelanomalyswap(node *root)
{
    vector<node*> rlo;
    stack<node*> s;
    queue<node*> q;
    node *p=root;
    q.push(p);
    while(!q.empty())
    {
        p=q.front();
        q.pop();
        s.push(p);
    }
}

```

```

    if(p->right)
    q.push(p->right);
    if(p->left)
    q.push(p->left);
}
while(!s.empty())
{
    rlo.push_back(s.top());
    s.pop();
}
return anomalyswap(rlo);
}
void huffmanadaptive(node *root,char ch)
{
    node *p=NULL;
    search(root,ch,p);
    if(p==NULL)
    {
        search(root,'#',p);
        p->data='$'; //different from hash
        p->left=newnode('#',0);
        p->right=newnode(ch,1);
    }
    else
    p->freq++;
    updatefreq(root);
    while(reverselevelanomalyswap(root))
    updatefreq(root);
    int h=height(root);
    int arr[h];
    printCodes(root,arr,0);
}
int main()
{
    cout<<"Enter running stream of characters"<<"\n";
    char ch,s;
    node *root=newnode('#',0);
    do
    {
        cin>>s;

```

```

huffmanadaptive(root,s);
cout<<"Do you want to continue(y/n): ";
cin>>ch;
}while(ch=='y'||ch=='Y');
return 0;
}

```

```

[Shivs-MacBook-Air:multimedia championballer$ g++ -std=c++11 prac4.cpp -o run
[Shivs-MacBook-Air:multimedia championballer$ ./run
Enter running stream of characters
h
# 0
h 1
Do you want to continue(y/n): y
e
# 0 0
e 0 1
h 1
Do you want to continue(y/n): y
l
h 0
# 1 0 0
l 1 0 1
e 1 1
Do you want to continue(y/n): y
l
l 0
# 1 0 0
h 1 0 1
e 1 1
Do you want to continue(y/n): y
o
l 0
e 1 0
# 1 1 0 0
o 1 1 0 1
h 1 1 1
Do you want to continue(y/n): n

```

# Practical 5

## WAP for text compression by Arithmetic coding in C

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    char c;
    double cummulative = 0.0, freq;
    cout << "How many characters are there\n";
    cin >> n;
    cout << "Enter char and its probability in alphabetical order\n";
    double range_from[256], range_to[256];
    memset(range_from, 0, sizeof range_from);
    memset(range_to, 0, sizeof range_to);
    for (int i = 0; i < n; i++)
    {
        cin >> c;
        cin >> freq;
        range_from[c] = cummulative;
        cummulative += freq;
        range_to[c] = cummulative;
        cout << range_from[c] << " - " << range_to[c] << endl;
    }
    string str;
    cout << "Enter string to be encoded" << endl;
    cin >> str;
    cout << endl;
    double low = 0.0, high = 1.0, diff = 1.0;
    n = str.length();
    for (int i = 0; i < n; i++)
    {
        high = low + (diff * range_to[str[i]]);
        low = low + (diff * range_from[str[i]]);
        diff = high - low;
    }
    double encoded = (low + high) / 2.0;
    cout << "Encoded string is " << encoded << endl;
```



```

cout << "Decoded string is ";
while (n--)
{
for (int i = 0; i < 256; i++) {
if (encoded >= range_from[i] && encoded < range_to[i])
{
c = i;
break;
}
}
cout << c;
encoded = (encoded - range_from[c]) / (range_to[c] -
range_from[c]);
}
return 0;
}

```

```

[Shivs-MacBook-Air:multimedia championballer$ g++ prac5.cpp -o run ]
[Shivs-MacBook-Air:multimedia championballer$ ./run ]
How many characters are there
5
Enter char and its probability in alphabetical order
e 0.2
0 - 0.2
c 0.2
0.2 - 0.4
h 0.1
0.4 - 0.5
o 0.3
0.5 - 0.8
. 0.2
0.8 - 1
Enter string to be encoded
echo.

Encoded string is 0.05908
Decoded string is echo.Shivs-MacBook-Air:multimedia championballer$ █

```

# Practical 6

## WAP for text compression by LZW compression technique in C

```
#include <bits/stdc++.h>

using namespace std;

vector<int> encryption(){
    string s;
    cin>>s;
    int N;
    map<string ,int>mapp;
    cout<<"Number of Distinct characters: ";
    cin>>N;
    while(N--){
        string ch;
        cin>>ch;
        mapp.insert({ch,mapp.size()+1});
    }
    int index;
    vector<int>v;
    int i=0;
    while(i<s.size()) {
        string str="";
        index=0;
        for(int j=i;j<s.size();j++)
        {
            str+=s[j];
            map<string ,int>::iterator it= mapp.find(str);
            if(it!=mapp.end()) {
                index=it->second;
                i++;
            }
            else {
                mapp.insert({str,mapp.size()+1});
                v.push_back(index);
                break;
            }
        }
    }
}
```

```

}
v.push_back(index);
auto it=mapp.begin();
while(it!=mapp.end()) {
    cout<<it->first<<" "<<it->second<<endl;
    it++;
}
for(int i=0;i<v.size();i++)
cout<<v[i]<<" ";
return v;
}
int main(){
encryption();
}

```

```

Shivs-MacBook-Air:multimedia championballer$ g++ -std=c++11 prac6.cpp -o run
Shivs-MacBook-Air:multimedia championballer$ ./run
waddawad
Number of Distinct characters: 3
w a d
a 2
ad 5
aw 8
d 3
da 7
dd 6
w 1
wa 4
wad 9
1 2 3 3 2 4 3 Shivs-MacBook-Air:multimedia championballer$

```

# Practical 7

## WAP to create a Keyboard Piano

HTML File :

```
<!DOCTYPE html>
<html>
<head>
  <title>Circles</title>
  <script type="text/javascript" src="paper-full.js"></script>
  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/
howler/1.1.28/howler.js"></script>
  <link rel="stylesheet" type="text/css" href="circles.css">

  <script type="text/paperscript" canvas="myCanvas">
    var keyData = {
      q: {
        sound: new Howl({
          urls: ['sounds/bubbles.mp3']
        }),
        color: '#1abc9c'
      },
      w: {
        sound: new Howl({
          urls: ['sounds/clay.mp3']
        }),
        color: '#2ecc71'
      },
      e: {
        sound: new Howl({
          urls: ['sounds/confetti.mp3']
        }),
        color: '#3498db'
      },
      r: {
        sound: new Howl({
          urls: ['sounds/corona.mp3']
        }),
        color: '#9b59b6'
      },
    },
```

```
t: {
  sound: new Howl({
    urls: ['sounds/dotted-spiral.mp3']
  }),
  color: '#34495e'
},
y: {
  sound: new Howl({
    urls: ['sounds/flash-1.mp3']
  }),
  color: '#16a085'
},
u: {
  sound: new Howl({
    urls: ['sounds/flash-2.mp3']
  }),
  color: '#27ae60'
},
i: {
  sound: new Howl({
    urls: ['sounds/flash-3.mp3']
  }),
  color: '#2980b9'
},
o: {
  sound: new Howl({
    urls: ['sounds/glimmer.mp3']
  }),
  color: '#8e44ad'
},
p: {
  sound: new Howl({
    urls: ['sounds/moon.mp3']
  }),
  color: '#2c3e50'
},
a: {
  sound: new Howl({
    urls: ['sounds/pinwheel.mp3']
  }),
```

```
        color: '#f1c40f'
    },
    s: {
        sound: new Howl({
            urls: ['sounds/piston-1.mp3']
        }),
        color: '#e67e22'
    },
    d: {
        sound: new Howl({
            urls: ['sounds/piston-2.mp3']
        }),
        color: '#e74c3c'
    },
    f: {
        sound: new Howl({
            urls: ['sounds/prism-1.mp3']
        }),
        color: '#95a5a6'
    },
    g: {
        sound: new Howl({
            urls: ['sounds/prism-2.mp3']
        }),
        color: '#f39c12'
    },
    h: {
        sound: new Howl({
            urls: ['sounds/prism-3.mp3']
        }),
        color: '#d35400'
    },
    j: {
        sound: new Howl({
            urls: ['sounds/splits.mp3']
        }),
        color: '#1abc9c'
    },
    k: {
        sound: new Howl({
```

```
        urls: ['sounds/squiggle.mp3']
      }),
      color: '#2ecc71'
    },
    l: {
      sound: new Howl({
        urls: ['sounds/strike.mp3']
      }),
      color: '#3498db'
    },
    z: {
      sound: new Howl({
        urls: ['sounds/suspension.mp3']
      }),
      color: '#9b59b6'
    },
    x: {
      sound: new Howl({
        urls: ['sounds/timer.mp3']
      }),
      color: '#34495e'
    },
    c: {
      sound: new Howl({
        urls: ['sounds/ufo.mp3']
      }),
      color: '#16a085'
    },
    v: {
      sound: new Howl({
        urls: ['sounds/veil.mp3']
      }),
      color: '#27ae60'
    },
    b: {
      sound: new Howl({
        urls: ['sounds/wipe.mp3']
      }),
      color: '#2980b9'
    },
  },
```

```

n: {
    sound: new Howl({
        urls: ['sounds/zig-zag.mp3']
    }),
    color: '#8e44ad'
},
m: {
    sound: new Howl({
        urls: ['sounds/moon.mp3']
    }),
    color: '#2c3e50'
}
}

```

```

var circles = [];

```

```

function onKeyDown(event) {
    if(keyData[event.key]){
        var maxPoint = new Point(view.size.width,
view.size.height);

        var randomPoint = Point.random();
        var point = maxPoint * randomPoint;
        var newCircle = new Path.Circle(point, 500)
        newCircle.fillColor = keyData[event.key].color;
        keyData[event.key].sound.play();
        circles.push(newCircle);
    }
}

```

```

function onFrame(event){
    for(var i = 0; i < circles.length; i++){
        circles[i].fillColor.hue += 1;
        circles[i].scale(.9);
    }
}

```

```

</script>

```

```

</head>

```

```

<body>

```

```

    <canvas id="myCanvas" resize></canvas>

```

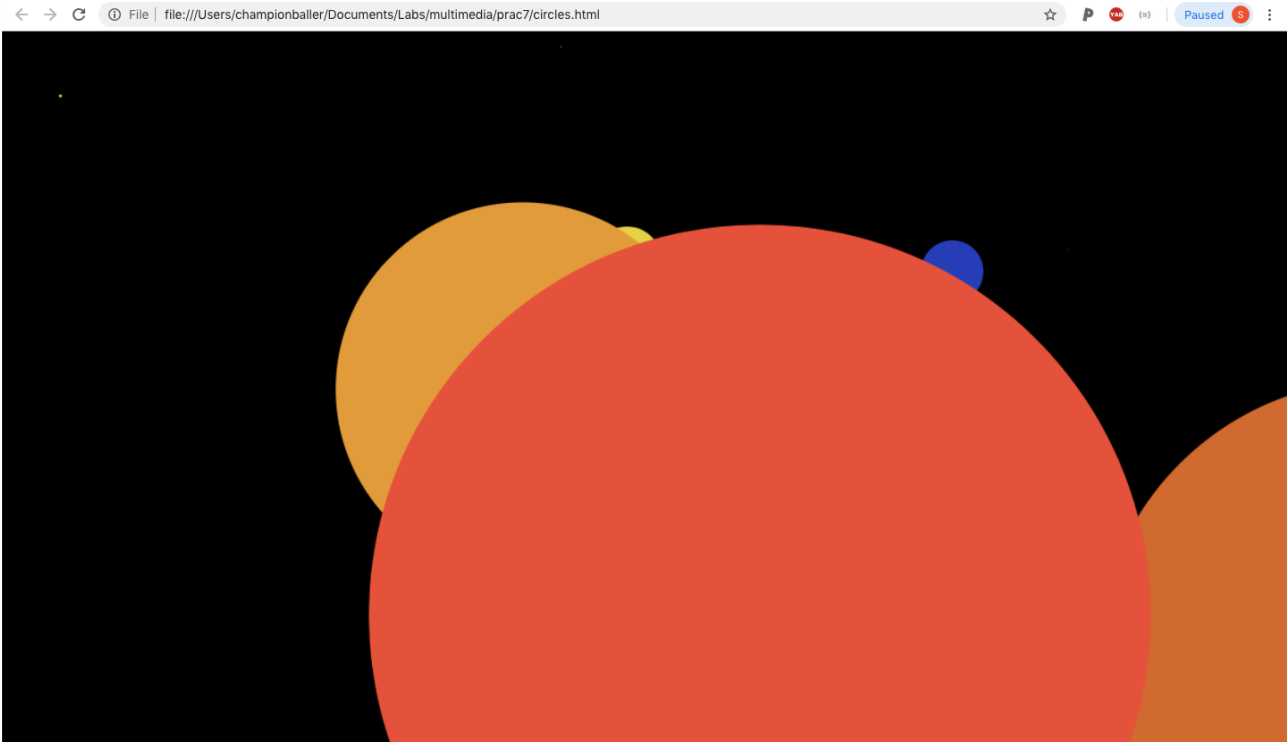
```

</body>

```



</html>



# Practical 8

**Create a web page to design your resume using basic HTML formatting Tags**

```
<div id="header"></div>
<div class="left"></div>
<div class="stuff">
  <br><br>
  <h1>Resume</h1>
  <h2>Shiv</h2>
  <hr />
  <br>
  <p class="head">Interests</p>
  <ul>
    <li>Drawing</li>
    <li>Photography</li>
    <li>Design</li>
    <li>Programming</li>
    <li>Computer Science</li>
  </ul>
  <p class="head">Skills</p>
  <ul>
    <li>Web Design with HTML & CSS</li>
  </ul>
  <p class="head">Education</p>
  <ul>
    <li>Manav Rachna International School</li>

    <li>Netaji Subhas Institute of Technology</li>
    <li>Codeacademy</li>
  </ul>
  <p class="head">Experience</p>
  <ul>
    <li>Undergraduate Researcher LCS2, IIITD</li>
    <li>Software development lead at ARES Robotics</li>
  </ul>
  <p class="head">Extracurriculars</p>
  <ul>
    <li>Recycling Club</li>
```

```
<li>Gardening Club</li>
<li>Book Club</li>
</ul>
</div>
<div class="right"></div>
<div id="footer">
```

# Resume

## Shiv

---

### Interests

- Drawing
- Photography
- Design
- Programming
- Computer Science

### Skills

- Web Design with HTML & CSS

### Education

- Manav Rachna International School
- Netaji Subhas Institute of Technology
- Codeacademy

### Experience

- Undergraduate Researcher LCS2, IIITD
- Software development lead at ARES Robotics

### Extracurriculars

- Recycling Club
- Gardening Club
- Book Club

# Practical 9

**Create a web page to design a time table of your college for all the semesters and branch using anchor tag**

```
<!DOCTYPE html>
<head>
  <!-- jQuery library -->
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></
script>
  <!-- Latest compiled JavaScript for materialize -->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.2/js/
materialize.min.js"></script>
  <!-- Latest compiled JavaScript for materialize -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
  <!-- fonts source of Droid serif Merriweather-->
  <link href="https://fonts.googleapis.com/css?family=Droid+Serifl
Merriweather:700" rel="stylesheet">
  <!--materialize css file-->
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.2/css/
materialize.min.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.cs
s">
  <!-- Materialize-Icons -->
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
</head>
<style>
.c {
background-color: blue;
width: 500px;
}
.b {
```

```
color: yellow;
font-size: 50px;
padding: 20px 0 0 40px;
font-family: Merriweather;
}
.a {
font-size: 20px;
padding: 10px 10px 10px 50px;
margin: 10px 0 0 90px;
width: 300px;
font-family: Merriweather;
background-color: peachpuff;
}
</style>
<html>
<div class="c">
<h1 class="b">Time Table NSIT</h1>
<div class="a"><a href="https://drive.google.com/open?
id=1BDf78lqSZVYlQYmtG98iPYYBhlvH9NSR">Time Table COE-1</a></div>
<div class="a"><a href="https://drive.google.com/open?
id=1PiARV7Qyxq7RrjLNGGcTcJFU1ilcg7eP">Time Table COE-2</a></div>
<div class="a"><a href="https://drive.google.com/open?
id=1MC6lyXXIXNA6EDMN0B6QDoOdN5kaEiq1">Time Table COE-3</a></
div>
<div class="a"><a href="https://drive.google.com/open?
id=1ABgvTxHjZ8C7Jp6cLLHVNKwEZ32oqy62">Time Table IT-1</a></div>
<div class="a"><a href="https://drive.google.com/open?
id=1IM_4gy9b6nt0tOtCD4JcSUhR_c13ezl4">Time Table IT-2</a></div>
<div class="a"><a href="">Time Table ECE-1</a></div>
<div class="a"><a href="">Time Table ECE-2</a></div>
<div class="a"><a href="">Time Table ECE-3</a></div>
<div class="a"><a href="">Time Table ICE-1</a></div>
<div class="a"><a href="">Time Table ICE-2</a></div>
<div class="a"><a href="">Time Table ICE-3</a></div>
<div class="a"><a href="">Time Table MPAE-1</a></div>
<div class="a"><a href="">Time Table MPAE-2</a></div>
<div class="a"><a href="">Time Table ME-1</a></div>
</div>
</html>
```

# Time Table NSIT

Time Table COE-1

Time Table COE-2

Time Table COE-3

Time Table IT-1

Time Table IT-2

Time Table ECE-1

Time Table ECE-2

Time Table ECE-3

Time Table ICE-1

Time Table ICE-2

Time Table ICE-3

Time Table MPAE-1

# Practical 10

## WAP to simulate the Sudoku puzzle in C

```
#include <bits/stdc++.h>
#define UNASSIGNED 0
#define N 9
using namespace std;
bool Find_empty(int grid[N][N], int &row, int &col)
{
    for (row = 0; row < N; row++)
        for (col = 0; col < N; col++)
            if (grid[row][col] == UNASSIGNED)
                return true;
    return false;
}
bool UsedInRow(int grid[N][N], int row, int num)
{
    for (int col = 0; col < N; col++)
        if (grid[row][col] == num)
            return true;
    return false;
}
bool UsedInCol(int grid[N][N], int col, int num)
{
    for (int row = 0; row < N; row++)
        if (grid[row][col] == num)
            return true;
    return false;
}
bool UsedInBox(int grid[N][N], int boxStartRow, int boxStartCol, int num)
{
    for (int row = 0; row < 3; row++)
        for (int col = 0; col < 3; col++)
            if (grid[row + boxStartRow][col + boxStartCol] == num)
                return true;
    return false;
}
bool isSafe(int grid[N][N], int row, int col, int num)
{

```

```

    return !UsedInRow(grid, row, num) && !UsedInCol(grid, col, num) && !
UsedInBox(grid, row - row % 3, col - col % 3, num);
}
bool SolveSudoku(int grid[N][N])
{
    int row, col;
    if (!Find_empty(grid, row, col))
        return true;
    for (int num = 1; num <= 9; num++)
    {
        if (isSafe(grid, row, col, num))
        {
            grid[row][col] = num;
            if (SolveSudoku(grid))
                return true;
            grid[row][col] = UNASSIGNED;
        }
    }
    return false;
}
void printGrid(int grid[N][N])
{
    for (int row = 0; row < N; row++)
    {
        for (int col = 0; col < N; col++)
            cout << grid[row][col] << " ";
        cout << endl;
    }
    cout << endl;
}
int main()
{
    int grid[N][N], temp[N][N];
    cout << "Enter the grid (0 for empty cells): " << endl;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            cin >> grid[i][j];
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            temp[i][j] = grid[i][j];

```



```

if (SolveSudoku(grid) == true)
{
char ch;
int r = 0, c = 0, v = 0;
if (!Find_empty(temp, r, c))
cout << "Grid Filled\n";
else
{
cout << endl
<< endl;
cout << "*****" << endl;
cout << " MAIN MENU " << endl;
cout << "*****" << endl;
cout << "1) Solve Sudoku" << endl;
cout << "2) See Solution" << endl;
cout << "*****" << endl;
cout << "Enter your choice (1/2): ";
cin >> ch;
if (ch == '1')
{
cout << "\nCurrent state of sudoku is: " << endl;
printGrid(temp);
ch = 'y';
while (Find_empty(temp, r, c) && (ch == 'y' || ch == 'Y'))
{
r = 0;
c = 0;
v = 0;
cout << "Enter 'Row Col value' for Input (e.g. 2 3 5):";
cin >> r >> c >> v;
if (temp[r][c] == 0)
{
if (grid[r][c] == v)
{
cout << "\nCorrect\n";
temp[r][c] = v;
}
else
{
cout << "\nWrong value, Try again\n";

```

```

}
}
else
cout << "\nBlock not empty\n";
cout << "\nCurrent state of sudoku is: " << endl;
printGrid(temp);
cout << "\nDo you want to continue (y/n): ";
cin >> ch;
}
}
cout << "\n\nCorrect solution for sudoku is: " << endl;
printGrid(grid);
}
}
else
printf("No solution exists");

return 0;
}

```

```

[Shivs-MacBook-Air:multimedia championballer$ g++ prac10.cpp -o run
[Shivs-MacBook-Air:multimedia championballer$ ./run

```

```

Enter the grid (0 for empty cells):

```

```

0 0 0 6 0 0 4 0 0
7 0 0 0 0 3 6 0 0
0 0 0 0 9 1 0 8 0
0 0 0 0 0 0 0 0 0
0 5 0 1 8 0 0 0 3
0 0 0 3 0 6 0 4 5
0 4 0 2 0 0 0 6 0
9 0 3 0 0 0 0 0 0
0 2 0 0 0 0 1 0 0

```

```

*****
MAIN MENU
*****

```

```

1) Solve Sudoku
2) See Solution
*****
Enter your choice (1/2): 1

```

```

Current state of sudoku is:

```

```

0 0 0 6 0 0 4 0 0
7 0 0 0 0 3 6 0 0
0 0 0 0 9 1 0 8 0
0 0 0 0 0 0 0 0 0
0 5 0 1 8 0 0 0 3
0 0 0 3 0 6 0 4 5
0 4 0 2 0 0 0 6 0
9 0 3 0 0 0 0 0 0
0 2 0 0 0 0 1 0 0

```

```

Enter 'Row Col value' for Input (e.g. 2 3 5):0 0 5

```

```

Correct

```

```

Current state of sudoku is:

```

```

5 0 0 6 0 0 4 0 0
7 0 0 0 0 3 6 0 0
0 0 0 0 9 1 0 8 0
0 0 0 0 0 0 0 0 0
0 5 0 1 8 0 0 0 3
0 0 0 3 0 6 0 4 5
0 4 0 2 0 0 0 6 0
9 0 3 0 0 0 0 0 0
0 2 0 0 0 0 1 0 0

```

```

Do you want to continue (y/n): y

```

```

Enter 'Row Col value' for Input (e.g. 2 3 5):0 1 3

```

```

Wrong value, Try again

```

```

Current state of sudoku is:

```

```

5 0 0 6 0 0 4 0 0
7 0 0 0 0 3 6 0 0
0 0 0 0 9 1 0 8 0
0 0 0 0 0 0 0 0 0
0 5 0 1 8 0 0 0 3
0 0 0 3 0 6 0 4 5
0 4 0 2 0 0 0 6 0
9 0 3 0 0 0 0 0 0

```