



**T.C.
ESKİŞEHİR OSMANGAZI UNIVERSITY
FACULTY OF ENGINEERING AND ARCHITECTURE
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER PROGRAMMING
FINAL EXAM – TAKE HOME**

		Grade
Answer (making program completely)	:	100
TOTAL	:	100

• **SORU**

Bu programda, **Bilbakalım** adlı bir kelime tahmin oyunu oluşturacaksınız. Bu oyunda, bilgisayar oluşturulmuş bir sözlükten (rastgele) gizli **beş harfli** bir kelime seçecektir. Oyuncu bir kelime tahmin edecek ve bu kelime sözlükte var ise bu tahmin geçerli bir tahmindir. Bir sözcük, yalnızca sözlükte bulunuyorsa ve aynı zamanda **beş farklı ayrı harften oluşuyorsa**, geçerli bir gizli kelimedir. Yaklaşık 2415 beş harfli kelimeden oluşan bir kelime listesi (**words.txt**) verilmiştir. Dosya her satıra bir tane kelime olmak üzere, tümü küçük harfli kelimelerden oluşmaktadır.

Dosyadan okunan tüm kelimeler `set<string>` `words` içinde tutulacak olup geçerli kelimeler `vector<string>` `secrets` içine akatarılacak ve bu vector içinden rastgele gizli kelime (**secret**) seçilecektir. Söz konusu bu yapılar ile ilgili örnekler aşağıdaki dosyalarda verilmiştir. Dikkatlice inceleyip kendi kodunuzda ilgili kısımları kullanabilirsiniz.

(bknz: Readfromfile.cpp, set_string_example.cpp, vector_string_example.cpp)

Her turda oyuncu bir kelime tahmin edecektir. Tahmin doğruysa, bilgisayar kaç tahminde bulunduğunu rapor edecek ve program sona erecektir. Tahmin yanlışsa, bilgisayar gizli kelime ve tahminde kaç harfin eşlendiğini ve bu harflerden kaçının doğru noktada olduğunu söyleyecektir. Oyuncu daha sonra bir sonraki tahminde bulunacaktır. Kullanıcı ipucu kodu 'xxxxx' girerse (ipucu kelimesi: gizli kelime) program gizli kelimeyi (**secret**) göstermelidir. İpucu kodunun kullanılması bir tahmin olarak sayılmalı ve programı sonlandırmamalıdır. Programınızın nasıl çalışması gerektiğini göstermek için örnek bir çalışma aşağıda verilmiştir;

Gizli kelimemizdeki herhangi bir harf yalnızca bir kez eşleştirilmelidir. Örneğin,

Gizli kelime: **cab**le

Tahmin : **ea**gle

4,3 değil 3,3 ile sonuçlanır.

Bu program iki sınıf kullanılarak kodlanmalıdır. (Ek olarak “ana fonksiyon” ve tercihen fanksiyon tanımlarının yapıldığı kaynak dosya).

- **Dictionary** adlı sınıf, kelimeleri girip saklamak, listeden rastgele bir kelime seçmek ve sözlükte olduklarını doğrulamak için kelimeleri arama işlevlerini gerçekleştirmelidir. **Dictionary** verimli arama için bir set <string> kullanacaktır. Rastgele bir sözcük seçen fonksiyon, yinelenen harfli bir sözcük seçmemelidir. (Tahminde ise tekrarlanan harfli kelime olabilir.)
- **Words** adlı diğer sınıf, oyun kurallarına dikkat edecek, oyun durumunu takip edecek ve kullanıcı giriş ve çıkışını yönetecektir.

Bu program birçok ayrıntı içeren bir programdır. Programınızı projenizin her aşamasında derlemenizi ve çalıştırmaya devam etmenizi tavsiye ediyorum. İlk başta tüm fonksiyonlarınızı derleme ve çalıştırmak için basit tutunuz.

Sınıflara ait örnek kodlar aşağıda verilmiştir. main fonksiyonu minimal kod ile çalışmalıdır.

```
class Words{
public:
    Words(){}; //Default Constructor
    ~Words() {}; //Destructor
    // public interface to play game of Words
    //it may call from main() using followings;
    //main(){
    //Words W;
    //W.playgame(W);
    //}
    void playgame(Words &);
private:
    // count functions compare string param with the
    // secret word
    // find common letters in any order
    int countofMatchLetters(string);
    // find common letters in place
    int countofTruePlaceLetters(string);
    // track number of guesses used
    int numguesses;
    // flag set to true if cheat code is used
    bool cheated;
    // word to guess
    string secret;
};
```

```
class Dictionary{
public:
    //Default Constructor
    Dictionary();
    //opens file and assign all words into
    set<string> words;
    Dictionary(char *filename);
    //Destructor
    ~Dictionary();

    // return a random word (without repeated
    // letters)
    string BringValidWord();
    // check if word is in Dictionary
    bool IsWordValid(string word);
private:
    // test if n. word has only unique letters
    bool TestLegalSecretWord(string word);
    set<string> words;
    vector<string> secrets;
};
```

Örnek Çıktı 1

Bilbakalım Oyununa Hosgeldiniz:
5 harften oluşan bir kelime tuttum.
 Tahmininiz Nedir? **xyxyz**
Bu kelime sozlukte yok!
 Tahmininiz Nedir? **hello**
 Eşleşen harf sayısı: **2**
 Eşleşen harflerin doğru konumu: **0**
 Tahmininiz Nedir? **stove**
 Eşleşen harf sayısı: **3**
 Eşleşen harflerin doğru konumu: **1**
 Tahmininiz Nedir? **venom**
 Eşleşen harf sayısı: **3**
 Eşleşen harflerin doğru konumu: **1**
 Tahmininiz Nedir? **verse**
 Eşleşen harf sayısı: **2**
 Eşleşen harflerin doğru konumu: **2**
 Tahmininiz Nedir? **vague**
 Eşleşen harf sayısı: **4**
 Eşleşen harflerin doğru konumu: **4**
 Tahmininiz Nedir? **vogue**
Bildiniz! 6 tahminde dogru sonuca ulaştınız!

Örnek Çıktı 2

Bilbakalım Oyununa Hosgeldiniz:
5 harften oluşan bir kelime tuttum.
 Tahmininiz Nedir? **draft**
 Eşleşen harf sayısı: **2**
 Eşleşen harflerin doğru konumu: **1**
 Tahmininiz Nedir? **xxxxx**
İpucu: Gizli Kelime: credo
 Tahmininiz Nedir? **greed**
 Eşleşen harf sayısı: **3**
 Eşleşen harflerin doğru konumu: **2**
 Tahmininiz Nedir? **credo**
Bildiniz! 4 tahminde ipucu kullanarak dogru sonuca ulaştınız!

WARNING!

YOUR PROGRAM WILL BE CHECKED FOR
 PLAGIARISM WITH A 3rd PARTY SOFTWARE.
 THEREFORE, YOU SHOULD DEVELOP YOUR
 ALGORITHM AND CODE COMPLETELY BY
 YOURSELF.