#include <stdio.h>

void getstr(char s[]);

int length(char s[]);

void copy(char s1[],char s2[]);

void reverse(char s1[]);

void palin(char s1[]);

int compare(char s1[],char s2[]);

void subString(char s1[],char s2[]);

int main()

{ char s1[20];char s2[20];int l,m,ch;

do

{ printf("Select the Options:\n");

printf("1. Length\n");

printf("2. Palindrome\n");

printf("3. String Compare\n");

printf("4. Copy\n");

printf("5. Reverse\n");

printf("6. Substring\n");

printf("7. Exit\n\n");

scanf("%d",&ch);

switch(ch)

{case 1:getstr(s1);

printf("String : %s\n",s1);

l=length(s1);

printf("Length of String : %d\n",l);

printf("\n\n"); break;

case 2: getstr(s1);

printf("String : %s\n",s1);

palin(s1);

printf("\n\n");

break;

case 3:getstr(s1);

printf("Enter a string to compare it with %s: ",s1);

scanf("%s",s2);

printf("String 1: %s\n",s1);

printf("String 2: %s\n",s2);

m=compare(s1,s2);

printf("Returned Value: %d\n\n",m);

break;

case 4:getstr(s1);

printf("Enter a string to copy it in %s: ",s1);

scanf("%s",s2);

printf("Before Copying:\n");

printf("String 1: %s\n",s1);

printf("String 2: %s\n\n",s2);

copy(s1,s2);

printf("After Copying:\n");

printf("String 1: %s\n",s1);

printf("String 2: %s\n\n",s2);

break;

case 5:getstr(s1);

printf("Before Reversing:\n");

printf("String : %s\n\n",s1); reverse(s1);

printf("After Reversing:\n");

printf("String : %s\n\n",s1);

break;

case 6:getstr(s1);

printf("Enter a string to check if it is a part of %s: ",s1);

scanf("%s",s2);

printf("String 1: %s\n",s1);

printf("String 2: %s\n",s2);

subString(s1,s2);

printf("\n\n");

break;

case 7:printf("END\n\n");

break;

default:printf("Enter Again/n");

}

}

while(ch!=7);

return 0;

}

void getstr(char s[])

{ printf("\nEnter string\n");

scanf("%s",s); }

int length(char s[])

{

int len=0;

int i=0;

while(s[i]!='\0')

{

len=len+1;

i++;

}

return len;

}

void copy(char s1[],char s2[])

{

int len=length(s2);

int i;

for(i=0;i<len;i++)

{

s1[i]=s2[i];

}

s1[i]='\0';

}

void reverse(char s1[])

{

int len=length(s1);

char s2[len];

copy(s2,s1);

int i,j=len-1;

for(i=0;i<len;i++)

{

s1[i]=s2[j--];

}

}

void palin(char s1[])

{

int len=length(s1);

char s2[len];

copy(s2,s1);

reverse(s2);

int i,flag=0;

for(i=0;i<len;i++)

{

if(s2[i]!=s1[i])

{

flag=1;

break;

}

}

if(flag==1)

{

printf("%s is not a palindrome string\n",s1);

}

else if(flag==0)

{

printf("%s is a palindrome string\n",s1);

}

}

int compare(char s1[],char s2[])

{

int i,flag=1;

int len1=length(s1);

int len2=length(s2);

int len;

if(len1>len2)

{ flag=0;

printf("\n String 2 is shorter\n");

}

else if(len1<len2)

{ flag=0;

printf("\n String 1 is shorter\n");

}

else

{ len=len1;

for(i=0;i<len;i++)

{

if(s1[i]!=s2[i])

{

flag=0;

return ((int)s1[i]-(int)s2[i]);

}

} }

if(flag==1)

{

return 0;

}

else {printf("\n Strings can't be compared\n");

return -1;

}

}

void subString(char s1[],char s2[])

{ int o=1;

int len1=length(s1);

int len2=length(s2);

int i,j=0,c=0,k=0;

for(i=0;i<len1-len2;i++)

{ for(j=0;j<len2;j++)

{

if(s1[i+j]!=s2[j])

{

break;

}

}

if(j==t)

{ printf("substring appears at %d position\n",i);

count++;

}

}

printf("substring appears %d times in string\n",count);

}

/\*Enter Again/nSelect the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

1

Enter string

happy

String : happy

Length of String : 5

Select the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

2

Enter string

noon

String : noon

noon is a palindrome string

Select the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

3

Enter string

happy

Enter a string to compare it with happy: sad

String 1: happy

String 2: sad

String 2 is shorter

Strings can't be compared

Returned Value: -1

Select the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

4

Enter string

happy

Enter a string to copy it in happy: sad

Before Copying:

String 1: happy

String 2: sad

After Copying:

String 1: sad

String 2: sad

Select the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

5

Enter string

happy

Before Reversing:

String : happy

After Reversing:

String : yppah

Select the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

Enter string

abcab

Enter a string to check if it is a part of abcab: ab

String 1: abcab

String 2: ab

ab is a part of abcab

Substring occurs at location 0

Substring occurs at location 3

no. of occurences 2

Select the Options:

1. Length

2. Palindrome

3. String Compare

4. Copy

5. Reverse

6. Substring

7. Exit

7

END \*/