|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2021\_1\_C++\_07\_개념활용 | 학번 : | 20185309 | 이름 : | 황명원 |

* 개념 활용

1. 디폴트 매개변수를 갖는 하나의 add() 함수를 작성하고 프로그램을 완성 하시오

int main() {

int a[] = {1,2,3,4,5};

int b[] = {6,7,8,9,10};

int c = add(a, 5); // 배열 a의 정수를 모두 더한 값 리턴

int d = add(a, 5, b); // 배열 a와 b의 정수를 모두 더한 값 리턴

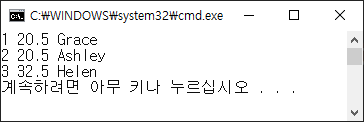
cout << c << endl; // 10 출력

cout << d << endl; // 55 출력

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  int add(int b[],int num1); int add(int b[],int num1,int c[]);  int main() {  int a[] = {1,2,3,4,5};  int b[] = {6,7,8,9,10};  int c = add(a, 5); // 배열 a의 정수를 모두 더한 값 리턴  int d = add(a, 5, b); // 배열 a와 b의 정수를 모두 더한 값 리턴  cout << c << endl; // 15 출력  cout << d << endl; // 55 출력 }  int add(int b[],int num1){  int total=0;  for(int i=0 ; i<num1 ; i++){  total+=b[i];  }  return total; } int add(int b[],int num1,int c[]){  int total=0;  for(int i=0 ; i<num1 ; i++){  total+=b[i]+c[i];  }  return total; } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. Person 클래스에 대하여 디폴트 생성자를 가진 하나의 생성자를 작성하고 프로그램을 완성하시오



class Person {

int id;

double weight;

string name;

public:

void show() const;

};

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class Person {  int id;  double weight;  string name; public:  Person(){  id=1;  weight=20.5;  name="Grace";  }  Person(int id,string name){  this->id=id;  weight=20.5;  this->name=name;  }  Person(int id,double weight,string name){  this->id=id;  this->weight=weight;  this->name=name;  }  ~Person();  void show() const; }; Person::~Person() {   }  void Person::show() const {  cout<<id<<" "<<weight<<" "<<name<<endl; }  int main(){  Person p1;  Person p2(2,"Ashley");  Person p3(3,32.5,"Helen");   p1.show();  p2.show();  p3.show(); } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 함수 big()을 호출하는 경우는 다음과 같다

int main() {

int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴

int y = big(300, 60); // 300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴

int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴

cout << x << ' ' << y << ' ' << z << endl;

}

(1) big() 함수를 2개 중복하여 작성하고 프로그램을 완성 하시오

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  int big(int num1,int num2); int big(int num1,int num2,int max);  int main() {  int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴  int y = big(300, 60); // 300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴  int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴  cout << x << ' ' << y << ' ' << z << endl; }  int big(int num1,int num2){  int num=100;  if(num1>num2){  if(num1>100)  return num;  else  return num1;  }  else{  if(num2>100)  return num;  else  return num2;  } } int big(int num1,int num2,int max){   if(num1>num2){  if(num1>max)  return max;  else  return num1;  }  else{  if(num2>max)  return max;  else  return num2;  } } |

ㅡ실행결과ㅡ

텍스트이(가) 표시된 사진

자동 생성된 설명

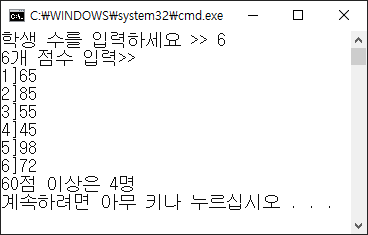
(2) 디폴트 매개변수를 가진 하나의 함수로 big()을 작성하고 프로그램을 완성 하시오

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;   int big(int num1,int num2,int max=100){   if(num1>num2){  if(num1>max)  return max;  else  return num1;  }  else{  if(num2>max)  return max;  else  return num2;  } }  int main() {  int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴  int y = big(300, 60); // 300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴  int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴  cout << x << ' ' << y << ' ' << z << endl; } |
| [실행결과] |

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 한 분반의 인원수를 입력 받아 성적이 60이상인 학생의 수를 계산하는 프로그램을 제시된 main()과 결과를 참고하여 완성 하시오.



class Dept {

int size; // scores 배열의 크기

int\* scores; // 동적 할당 받을 정수 배열의 주소

public:

Dept(int size); //매개변수로 받은 배열 크기만큼 scores 배열 생성

Dept(Dept& dept); //복사 생성자

~Dept(); // 소멸자

int getSize(); //배열 크기 반환

void read(); // size 만큼 키보드에서 정수를 읽어 scores 배열에 저장

bool isOver60(int index); // index의 학생의 성적이 60보다 크면 true 리턴

};

int main() {

Dept \*com;

int cnt;

cout << "학생 수를 입력하세요 >> ";

cin >> cnt;

com = new Dept(cnt); //입력한 학생 수만큼 scores 배열 생성

com->read(); //학생들의 성적을 키보드로부터 읽어 scores 배열에 저장

int n = **countPass(\*com);** // com 학과에 60점 이상으로 통과한 학생의 수를 리턴

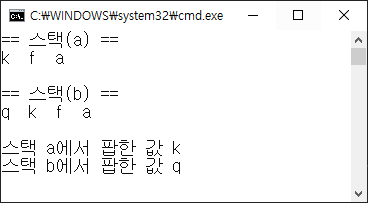
cout << "60점 이상은 " << n << "명" << endl;

delete com;

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  class Dept {  int size; // scores 배열의 크기  int\* scores; // 동적 할당 받을 정수 배열의 주소 public:  Dept(int size){  this->size=size;  scores=new int[size];  } //매개변수로 받은 배열 크기만큼 scores 배열 생성  Dept(Dept& dept){  this->size=dept.size;  } //복사 생성자  ~Dept(){  delete scores;  } // 소멸자  int getSize(){  return size;  } //배열 크기 반환  void read(); // size 만큼 키보드에서 정수를 읽어 scores 배열에 저장  bool isOver60(int index); // index의 학생의 성적이 60보다 크면 true 리턴 };  void Dept::read() {  cout<<size<<"개 점수 입력>>"<<endl;  for(int i=0 ; i<size ;i++){  cout<<i+1<<" ] ";  cin>>scores[i];  } } bool Dept::isOver60(int index) {  if(scores[index]>=60)  return true;  else  return false; }   int countPass(Dept &c){  int total=0;  for(int i=0 ; i<c.getSize() ; i++){  if(c.isOver60(i)==true)  total+=1;  }  return total; }  int main() {  Dept \*com;  int cnt;  cout << "학생 수를 입력하세요 >> ";  cin >> cnt;  com = new Dept(cnt); //입력한 학생 수만큼 scores 배열 생성  com->read(); //학생들의 성적을 키보드로부터 읽어 scores 배열에 저장  int n = countPass(\*com); // com 학과에 60점 이상으로 통과한 학생의 수를 리턴  cout << "60점 이상은 " << n << "명" << endl;  delete com; } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 문자를 저장할 수 있는 스택 클래스 MyInStack을 구현하고 제시된 결과처럼 실행될 수 있도록 프로그램을 완성 하시오. 단, 스택에 저장할 수 있는 최대 문자수는 생성자매개변수로 주어지고 size멤버에 저장한다.



class MyIntStack {

char \*p; // 스택 메모리로 사용할 포인터

int size; // 스택의 최대 크기

int tos; // 스택의 탑을 가리키는 인덱스

public:

MyIntStack();

MyIntStack(int size);

MyIntStack(MyIntStack& s);

~MyIntStack();

bool push(char n); // 정수 n을 스택에 푸시,스택이 꽉 차 있으면 false를, 아니면 true 리턴

bool pop(char &n); // 스택의 탑에 있는 값을 n에 팝, 만일 스택이 비어 있으면 false를, 아니면 true 리턴

void show(); //스택에 저장된 모든 문자 출력

};

int main() {

MyIntStack a(20);

a.push('a');

a.push('f');

a.push('k');

cout << "== 스택(a) ==" << endl;

a.show();

MyIntStack b = a; // 복사 생성

cout << endl<<"== 스택(b) ==" << endl;

b.push('q');

b.show();

char n;

a.pop(n); // 스택 a 팝

cout << endl<<"스택 a에서 팝한 값 " << n << endl;

b.pop(n); // 스택 b 팝

cout << "스택 b에서 팝한 값 " << n << endl<<endl;

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  class MyIntStack {  char \*p; // 스택 메모리로 사용할 포인터  int size; // 스택의 최대 크기  int tos; // 스택의 탑을 가리키는 인덱스 public:  MyIntStack(){  size=0;  p=new char[size];  tos=0;  }  MyIntStack(int size){  this->size=size;  p=new char[size];  tos=0;  }  MyIntStack(MyIntStack& s){  this->size=s.size;  this->p=new char[s.size];  this->tos=s.tos;  for(int i=0 ; i<=s.tos ; i++)  this->p[i]=s.p[i];  }  ~MyIntStack(){  delete []p;  }   bool push(char n); // 정수 n을 스택에 푸시,스택이 꽉 차 있으면 false를, 아니면 true 리턴  bool pop(char &n); // 스택의 탑에 있는 값을 n에 팝, 만일 스택이 비어 있으면 false를, 아니면 true 리턴  void show(); //스택에 저장된 모든 문자 출력 }; bool MyIntStack::push(char n) {  if(tos==size) {  cout<<"최대값을 넘었습니다."<<endl;  return false;  }  else {  p[tos] = n;  tos++;  return true;  } } bool MyIntStack::pop(char &n) {  if(tos==-1) {  cout<<"더이상 팝할 값이 없습니다."<<endl;  return false;  }  else{  p[tos]=n;  tos--;  return true;  } } void MyIntStack::show() {  for(int i=tos-1 ; i>=0 ; i--){  cout<<p[i]<<" ";  }  cout<<endl; }   int main() {  MyIntStack a(20);  a.push('a');  a.push('f');  a.push('k');   cout << "== 스택(a) ==" << endl;  a.show();   MyIntStack b = a; // 복사 생성  cout << endl<<"== 스택(b) ==" << endl;  b.push('q');  b.show();   char n='k';  a.pop(n); // 스택 a 팝  cout << endl<<"스택 a에서 팝한 값 " << n << endl;  n='q';  b.pop(n); // 스택 b 팝  cout << "스택 b에서 팝한 값 " << n << endl<<endl; } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |